Texas Instruments

PCI1410A GHK/GGU/PGE

PC Card Controllers

Data Manual

2000

PCIBus Solutions

SCPS057

IMPORTANT NOTICE

Texas Instruments and its subsidiaries (TI) reserve the right to make changes to their products or to discontinue any product or service without notice, and advise customers to obtain the latest version of relevant information to verify, before placing orders, that information being relied on is current and complete. All products are sold subject to the terms and conditions of sale supplied at the time of order acknowledgment, including those pertaining to warranty, patent infringement, and limitation of liability.

TI warrants performance of its semiconductor products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are utilized to the extent TI deems necessary to support this warranty. Specific testing of all parameters of each device is not necessarily performed, except those mandated by government requirements.

Customers are responsible for their applications using TI components.

In order to minimize risks associated with the customer's applications, adequate design and operating safeguards must be provided by the customer to minimize inherent or procedural hazards.

TI assumes no liability for applications assistance or customer product design. TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such semiconductor products or services might be or are used. TI's publication of information regarding any third party's products or services does not constitute TI's approval, warranty or endorsement thereof.

Contents

Section *Title* *Page*

1 Introduction 1–1
1.1 Description 1–1
1.2 Features 1–1
1.3 Related Documents 1–2
1.4 Trademarks 1–2
1.5 Ordering Information 1–2
2 Terminal Descriptions 2–1
3 Feature/Protocol Descriptions 3–1
3.1 Power-Supply Sequencing 3–1
3.2 I/O Characteristics 3–2
3.3 Clamping Voltages 3–2
3.4 Peripheral Component Interconnect (PCI) Interface 3–2
3.4.1 PCI Bus Lock ($\overline{\text{LOCK}}$) 3–2
3.4.2 Loading Subsystem Identification 3–3
3.5 PC Card Applications 3–3
3.5.1 PC Card Insertion/Removal and Recognition 3–3
3.5.2 P^2C Power-Switch Interface (TPS2211) 3–4
3.5.3 Zoomed-Video Support 3–5
3.5.4 Ultrazoomed Video 3–6
3.5.5 Internal Ring Oscillator 3–6
3.5.6 Integrated Pullup Resistors for PC Card Interface 3–7
3.5.7 SPKROUT and CAUDPWM Usage 3–7
3.5.8 LED Socket Activity Indicators 3–8
3.5.9 PC Card-16 Distributed DMA Support 3–8
3.5.10 PC Card-16 PC/PCI DMA 3–10
3.5.11 CardBus Socket Registers 3–10
3.6 Serial Bus Interface 3–11
3.6.1 Serial Bus-Interface Implementation 3–11
3.6.2 Serial Bus-Interface Protocol 3–11
3.6.3 Serial Bus EEPROM Application 3–13
3.6.4 Accessing Serial Bus Devices Through Software 3–14
3.7 Programmable Interrupt Subsystem 3–15
3.7.1 PC Card Functional and Card Status Change Interrupts . 3–15
3.7.2 Interrupt Masks and Flags 3–16
3.7.3 Using Parallel IRQ Interrupts 3–17
3.7.4 Using Parallel PCI Interrupts 3–18

3.7.5 Using Serialized IRQSER Interrupts 3–18
3.7.6 SMI Support in the PCI1410A Device 3–18
3.8 Power-Management Overview 3–19
3.8.1 Clock-Run Protocol 3–19
3.8.2 CardBus PC Card Power Management 3–19
3.8.3 16-Bit PC Card Power Management 3–19
3.8.4 Suspend Mode 3–20
3.8.5 Requirements for Suspend Mode 3–21
3.8.6 Ring Indicate 3–21
3.8.7 PCI Power Management 3–22
3.8.8 CardBus Device Class Power Management 3–23
3.8.9 ACPI Support 3–23
3.8.10 Master List of $\overline{\text{PME}}$ Context Bits and Global Reset-Only Bits 3–24

4 PC Card Controller Programming Model 4–1

4.1 PCI Configuration Registers 4–1
4.2 Vendor ID Register 4–2
4.3 Device ID Register 4–2
4.4 Command Register 4–3
4.5 Status Register 4–4
4.6 Revision ID Register 4–5
4.7 PCI Class Code Register 4–5
4.8 Cache Line Size Register 4–5
4.9 Latency Timer Register 4–6
4.10 Header Type Register 4–6
4.11 BIST Register 4–6
4.12 CardBus Socket/ExCA Base-Address Register 4–7
4.13 Capability Pointer Register 4–7
4.14 Secondary Status Register 4–8
4.15 PCI Bus Number Register 4–9
4.16 CardBus Bus Number Register 4–9
4.17 Subordinate Bus Number Register 4–9
4.18 CardBus Latency Timer Register 4–10
4.19 Memory Base Registers 0, 1 4–10
4.20 Memory Limit Registers 0, 1 4–11
4.21 I/O Base Registers 0, 1 4–11
4.22 I/O Limit Registers 0, 1 4–12
4.23 Interrupt Line Register 4–12
4.24 Interrupt Pin Register 4–13
4.25 Bridge Control Register 4–14
4.26 Subsystem Vendor ID Register 4–15
4.27 Subsystem ID Register 4–15
4.28 PC Card 16-Bit I/F Legacy-Mode Base-Address Register 4–15
4.29 System Control Register 4–16

4.30 Multifunction Routing Register 4–19
4.31 Retry Status Register 4–20
4.32 Card Control Register 4–21
4.33 Device Control Register 4–22
4.34 Diagnostic Register 4–23
4.35 Socket DMA Register 0 4–24
4.36 Socket DMA Register 1 4–25
4.37 Capability ID Register 4–26
4.38 Next-Item Pointer Register 4–26
4.39 Power Management Capabilities Register 4–27
4.40 Power Management Control/Status Register 4–28
4.41 Power Management Control/Status Bridge Support Extensions Register 4–29
4.42 Power Management Data Register 4–29
4.43 General-Purpose Event Status Register 4–30
4.44 General-Purpose Event Enable Register 4–31
4.45 General-Purpose Input Register 4–32
4.46 General-Purpose Output Register 4–33
4.47 Serial Bus Data Register 4–33
4.48 Serial Bus Index Register 4–34
4.49 Serial Bus Slave Address Register 4–34
4.50 Serial Bus Control and Status Register 4–35
5 ExCA Compatibility Registers 5–1
5.1 ExCA Identification and Revision Register 5–4
5.2 ExCA Interface Status Register 5–5
5.3 ExCA Power Control Register 5–6
5.3.1 Intel 82365SL Support 5–6
5.3.2 Intel 82365SL-DF Support 5–7
5.4 ExCA Interrupt and General Control Register 5–8
5.5 ExCA Card Status-Change Register 5–9
5.6 ExCA Card Status-Change-Interrupt Configuration Register 5–10
5.7 ExCA Address Window Enable Register 5–11
5.8 ExCA I/O Window Control Register 5–12
5.9 ExCA I/O Windows 0 and 1 Start-Address Low-Byte Registers 5–13
5.10 ExCA I/O Windows 0 and 1 Start-Address High-Byte Registers 5–13
5.11 ExCA I/O Windows 0 and 1 End-Address Low-Byte Registers 5–14
5.12 ExCA I/O Windows 0 and 1 End-Address High-Byte Registers 5–14
5.13 ExCA Memory Windows 0–4 Start-Address Low-Byte Registers 5–15
5.14 ExCA Memory Windows 0–4 Start-Address High-Byte Registers 5–16
5.15 ExCA Memory Windows 0–4 End-Address Low-Byte Registers 5–17
5.16 ExCA Memory Windows 0–4 End-Address High-Byte Registers 5–18
5.17 ExCA Memory Windows 0–4 Offset-Address Low-Byte Registers 5–19
5.18 ExCA Memory Windows 0–4 Offset-Address High-Byte Registers 5–20
5.19 ExCA Card Detect and General Control Register 5–21

5.20 ExCA Global Control Register 5–22
5.21 ExCA I/O Windows 0 and 1 Offset-Address Low-Byte Registers . . . 5–23
5.22 ExCA I/O Windows 0 and 1 Offset-Address High-Byte Registers . . . 5–23
5.23 ExCA Memory Windows 0–4 Page Register 5–24

6 CardBus Socket Registers 6–1

6.1 Socket Event Register 6–2
6.2 Socket Mask Register 6–3
6.3 Socket Present State Register 6–4
6.4 Socket Force Event Register 6–6
6.5 Socket Control Register 6–7
6.6 Socket Power Management Register 6–8

7 Distributed DMA (DDMA) Registers 7–1

7.1 DMA Current Address/Base Address Register 7–1
7.2 DMA Page Register 7–2
7.3 DMA Current Count/Base Count Register 7–2
7.4 DMA Command Register 7–3
7.5 DMA Status Register 7–3
7.6 DMA Request Register 7–4
7.7 DMA Mode Register 7–4
7.8 DMA Master Clear Register 7–5
7.9 DMA Multichannel/Mask Register 7–5

8 Electrical Characteristics 8–1

8.1 Absolute Maximum Ratings Over Operating Temperature Ranges 8–1
8.2 Recommended Operating Conditions 8–2
8.3 Electrical Characteristics Over Recommended Operating Conditions 8–3
8.4 PCI Clock/Reset Timing Requirements Over Recommended Ranges of Supply Voltage and Operating Free-Air Temperature . . . 8–4
8.5 PCI Timing Requirements Over Recommended Ranges of Supply Voltage and Operating Free-Air Temperature 8–4
8.6 PC Card Cycle Timing 8–4

9 Mechanical Information 9–1

List of Illustrations

Figure	Title	Page
2–1	PCI-to-CardBus Terminal Diagram	2–1
2–2	PCI-to-PC Card (16-Bit) Terminal Diagram	2–2
2–3	GGU Package Terminal Diagram	2–3
2–4	GHK Package Terminal Diagram	2–3
3–1	PCI1410A Simplified Block Diagram	3–1
3–2	3-State Bidirectional Buffer	3–2
3–3	TPS2211 Terminal Assignments	3–4
3–4	TPS2211 Typical Application	3–5
3–5	Zoomed-Video Implementation Using the PCI1410A Device	3–5
3–6	Zoomed-Video Switching Application	3–6
3–7	Sample Application of SPKROUT and CAUDPWM	3–8
3–8	Two Sample LED Circuits	3–8
3–9	Serial EEPROM Application	3–11
3–10	Serial Bus Start/Stop Conditions and Bit Transfers	3–12
3–11	Serial Bus-Protocol Acknowledge	3–12
3–12	Serial Bus Protocol – Byte Write	3–13
3–13	Serial Bus Protocol – Byte Read	3–13
3–14	EEPROM Interface Doubleword Data Collection	3–13
3–15	EEPROM Data Format	3–14
3–16	IRQ Implementation	3–18
3–17	Suspend Logic Diagram	3–20
3–18	Signal Diagram of Suspend Function	3–21
3–19	$\overline{\text{RI_OUT}}$ Functional Diagram	3–22
3–20	Block Diagram of a Status/Enable Cell	3–24
5–1	ExCA Register Access Through I/O	5–1
5–2	ExCA Register Access Through Memory	5–1
6–1	Accessing CardBus Socket Registers Through PCI Memory	6–1

List of Tables

Table *Title* *Page*

2–1 CardBus and 16-Bit PC Card Signal Names by GGU Terminal Number . . 2–4
2–2 CardBus and 16-Bit PC Card Signal Names by PGE Terminal Number . . . 2–5
2–3 CardBus and 16-Bit PC Card Signal Names by GHK Terminal Number . . . 2–6
2–4 CardBus PC Card Signal Names Sorted Alphabetically to GGU/PGE/GHK Terminal Number 2–8
2–5 16-Bit PC Card Signal Names Sorted Alphabetically to GGU/PGE/GHK Terminal Number 2–10
2–6 Power Supply Terminals 2–12
2–7 PC Card Power-Switch Terminals 2–12
2–8 PCI System Terminals 2–12
2–9 PCI Address and Data Terminals 2–13
2–10 PCI Interface Control Terminals 2–14
2–11 Multifunction and Miscellaneous Terminals 2–15
2–12 16-Bit PC Card Address and Data Terminals 2–16
2–13 16-Bit PC Card Interface Control Terminals 2–17
2–14 CardBus PC Card Interface System Terminals 2–18
2–15 CardBus PC Card Address and Data Terminals 2–19
2–16 CardBus PC Card Interface Control Terminals 2–20
3–1 PC Card Card-Detect and Voltage-Sense Connections 3–4
3–2 Distributed DMA Registers 3–9
3–3 PC/PCI Channel Assignments 3–10
3–4 I/O Addresses Used for PC/PCI DMA 3–10
3–5 CardBus Socket Registers 3–11
3–6 Registers and Bits Loadable Through Serial EEPROM 3–13
3–7 PCI1410A Registers Used to Program Serial Bus Devices 3–15
3–8 PC Card Interrupt Events and Description 3–16
3–9 Interrupt Mask and Flag Registers 3–17
3–10 SMI Control 3–18
3–11 Power-Management Registers 3–23
4–1 PCI Configuration Registers 4–1
4–2 Bit-Field Access Tag Descriptions 4–2
4–3 Command Register Description 4–3
4–4 Status Register Description 4–4
4–5 Secondary Status Register Description 4–8
4–6 Bridge Control Register Description 4–14
4–7 System Control Register Description 4–17
4–8 Multifunction Routing Register Description 4–19
4–9 Retry Status Register Description 4–20

4–10 Card Control Register Description 4–21
4–11 Device Control Register Description 4–22
4–12 Diagnostic Register Description 4–23
4–13 Socket DMA Register 0 Description 4–24
4–14 Socket DMA Register 1 Description 4–25
4–15 Power Management Capabilities Register Description 4–27
4–16 Power Management Control/Status Register Description 4–28
4–17 Power Management Control/Status Bridge Support Extensions Register Description 4–29
4–18 General-Purpose Event Status Register Description 4–30
4–19 General-Purpose Event Enable Register Description 4–31
4–20 General-Purpose Input Register Description 4–32
4–21 General-Purpose Output Register Description 4–33
4–22 Serial Bus Data Register Description 4–33
4–23 Serial Bus Index Register Description 4–34
4–24 Serial Bus Slave Address Register Description 4–34
4–25 Serial Bus Control and Status Register Description 4–35
5–1 ExCA Registers and Offsets 5–2
5–2 ExCA Identification and Revision Register Description 5–4
5–3 ExCA Interface Status Register Description 5–5
5–4 ExCA Power Control Register 82365SL Support Description 5–6
5–5 ExCA Power Control Register 82365SL-DF Support Description 5–7
5–6 ExCA Interrupt and General Control Register Description 5–8
5–7 ExCA Card Status-Change Register Description 5–9
5–8 ExCA Card Status-Change-Interrupt Configuration Register Description . . 5–10
5–9 ExCA Address Window Enable Register Description 5–11
5–10 ExCA I/O Window Control Register Description 5–12
5–11 ExCA Memory Windows 0–4 Start-Address High-Byte Registers Description 5–16
5–12 ExCA Memory Windows 0–4 End-Address High-Byte Registers Description 5–18
5–13 ExCA Memory Windows 0–4 Offset-Address High-Byte Registers Description 5–20
5–14 ExCA Card Detect and General Control Register Description 5–21
5–15 ExCA Global Control Register Description 5–22
6–1 CardBus Socket Registers 6–1
6–2 Socket Event Register Description 6–2
6–3 Socket Mask Register Description 6–3
6–4 Socket Present State Register Description 6–4
6–5 Socket Force Event Register Description 6–6
6–6 Socket Control Register Description 6–7
6–7 Socket Power Management Register Description 6–8

7–1 Distributed DMA Registers 7–1
7–2 DMA Command Register Description 7–3
7–3 DMA Status Register Description 7–3
7–4 DMA Mode Register Description 7–4
7–5 DMA Multichannel/Mask Register Description 7–5
8–1 PC Card Address Setup Time, $t_{su(A)}$, 8-Bit and 16-Bit PCI Cycles 8–4
8–2 PC Card Command Active Cycle Time, $t_{c(A)}$, 8-Bit PCI Cycles 8–5
8–3 PC Card Command Active Cycle Time, $t_{c(A)}$, 16-Bit PCI Cycles 8–5
8–4 PC Card Address Hold Time, $t_{h(A)}$, 8-Bit and 16-Bit PCI Cycles 8–5

1 Introduction

1.1 Description

The TI™ PCI1410A device is a high-performance PCI-to-PC Card controller that supports a single PC Card socket compliant with the *PC Card Standard*. The PCI1410A device provides features that make it the best choice for bridging between PCI and PC Cards in both notebook and desktop computers. The *PC Card Standard* retains the 16-bit PC Card specification defined in *PCI Local Bus Specification* and defines the new 32-bit PC Card, CardBus, as being capable of full 32-bit data transfers at 33 MHz. The PCI1410A device supports both 16-bit and CardBus PC Cards, powered at 5 V or 3.3 V, as required.

The PCI1410A device is compliant with the *PCI Local Bus Specification*, and its PCI interface can act as either a PCI master device or a PCI slave device. The PCI bus mastering is initiated during 16-bit PC Card DMA transfers or CardBus PC Card bridging transactions. The PCI1410A device also is compliant with the latest *PCI Bus Power Management Interface Specification* and *PCI Bus Power Management Interface Specification for PCI to CardBus Bridges*.

All card signals are buffered internally to allow hot insertion and removal without external buffering. The PCI1410A device is register-compatible with the Intel™ 82365SL-DF and 82365SL ExCA controllers. The PCI1410A internal data-path logic allows the host to access 8-, 16-, and 32-bit cards using full 32-bit PCI cycles for maximum performance. Independent buffering and a pipeline architecture provide an unsurpassed performance level with sustained bursting. The PCI1410A device also can be programmed to accept fast-posted writes to improve system-bus utilization.

Multiple system-interrupt signaling options are provided, including: parallel PCI, parallel ISA, serialized ISA, and serialized PCI. Furthermore, general-purpose inputs and outputs are provided for the board designer to implement sideband functions. Many other features designed into the PCI1410A device, such as socket-activity light-emitting diode (LED) outputs, are discussed in detail throughout the design specification.

An advanced complementary metal-oxide semiconductor (CMOS) process achieves low system power consumption, while operating at PCI clock rates up to 33 MHz. Several low-power modes enable the host power management system to further reduce power consumption.

1.2 Features

The PCI1410A device supports the following features:

- Ability to wake from $D3_{hot}$ and $D3_{cold}$
- Full compatiblity with the Intel 430TX (Mobile Triton II) chipset
- A 144-terminal low-profile QFP (PGE), 144-terminal MicroStar BGA™ ball grid array (GGU) package, or 209-terminal MicroStar BGA™ (GHK) package
- 3.3-V core logic with universal PCI interfaces compatible with 3.3-V and 5-V PCI signaling environments
- Mix-and-match 5-V/3.3-V 16-bit PC Cards and 3.3-V CardBus Cards
- Single PC Card or CardBus slot with hot insertion and removal
- Burst transfers to maximize data throughput on the PCI bus and the CardBus bus
- Parallel PCI interrupts, parallel ISA IRQ and parallel PCI interrupts, serial ISA IRQ with parallel PCI interrupts, and serial ISA IRQ and PCI interrupts
- Serial EEPROM interface for loading subsystem ID and subsystem vendor ID
- Pipelined architecture allows greater than 130 Mbit/s sustained throughput from CardBus to PCI and from PCI to CardBus

- Interface to parallel single-slot PC Card power-switch interfaces like the TI™ TPS2211 device
- Up to five general-purpose I/Os
- Programmable output select for $\overline{\text{CLKRUN}}$
- Five PCI memory windows and two I/O windows available to the 16-bit PC Card socket
- Two I/O windows and two memory windows available to the CardBus socket
- Exchangeable card architecture (ExCA) compatible registers are mapped in memory and I/O space
- Compatibility with Intel 82365SL-DF and 82365SL registers
- Distributed DMA (DDMA) and PC/PCI DMA
- 16-bit DMA on the PC Card socket
- Ring indicate, $\overline{\text{SUSPEND}}$, PCI $\overline{\text{CLKRUN}}$, and CardBus $\overline{\text{CCLKRUN}}$
- Socket-activity LED pins
- PCI bus lock ($\overline{\text{LOCK}}$)
- Advanced submicron, low-power CMOS technology
- Internal ring oscillator

1.3 Related Documents

- *Advanced Configuration and Power Interface (ACPI) Specification* (Revision 2.0)
- *PCI Bus Power Management Interface Specification* (Revision 1.1)
- *PCI Bus Power Management Interface Specification for PCI to CardBus Bridges* (Revision 0.6)
- *PCI Local Bus Specification* (Revision 2.2)
- *PCI Mobile Design Guide* (Revision 1.0)
- *PCI14xx Implementation Guide for D3 Wake-Up*
- *PC Card Standard*, Release 7
- *PC 98*
- *PC 99*
- *Serialized IRQ Support for PCI Systems* (Revision 6)

1.4 Trademarks

MicroStar BGA and TI are trademarks of Texas Instruments.

Intel is a trademark of Intel Corporation.

Maxim is a trademark of Maxim Integrated Products, Inc.

Other trademarks are the property of their respective owners.

1.5 Ordering Information

ORDERING NUMBER	NAME	VOLTAGE	PACKAGE
PCI1410A	PC Card controller	3.3-V, 5-V tolerant I/Os	144-terminal LQFP 144-terminal PBGA 209-terminal PBGA

2 Terminal Descriptions

The PCI1410A device is packaged in either a 144-terminal GGU MicroStar BGA™ or a 144-terminal PGE package. It also is packaged in a 209-terminal GHK MicroStar BGA™ that is pin-compatible with the TI™ PCI4410A device. The PCI4410A device is a single-socket CardBus bridge with integrated OHCI link. Figure 2–1 is a PGE-package terminal diagram showing PCI-to-CardBus signal names. Figure 2–2 is a PGE-package terminal diagram showing PCI-to-PC Card signal names. Figure 2–3 and Figure 2–4 are terminal diagrams for the GGU and GHK packages, respectively.

PGE LOW-PROFILE QUAD FLAT PACKAGE
(TOP VIEW)

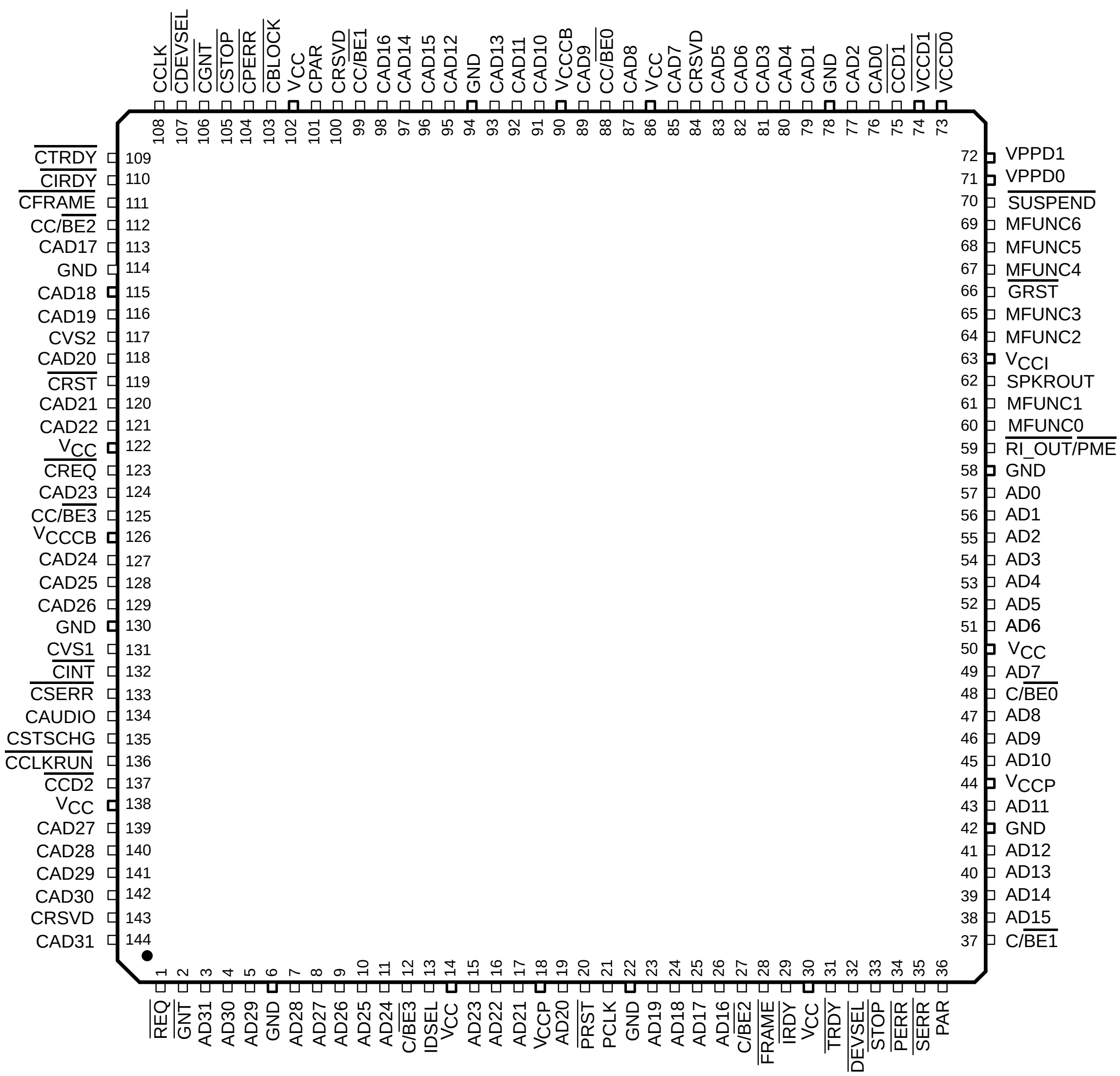


Figure 2–1. PCI-to-CardBus Terminal Diagram

**PGE LOW-PROFILE QUAD FLAT PACKAGE
(TOP VIEW)**

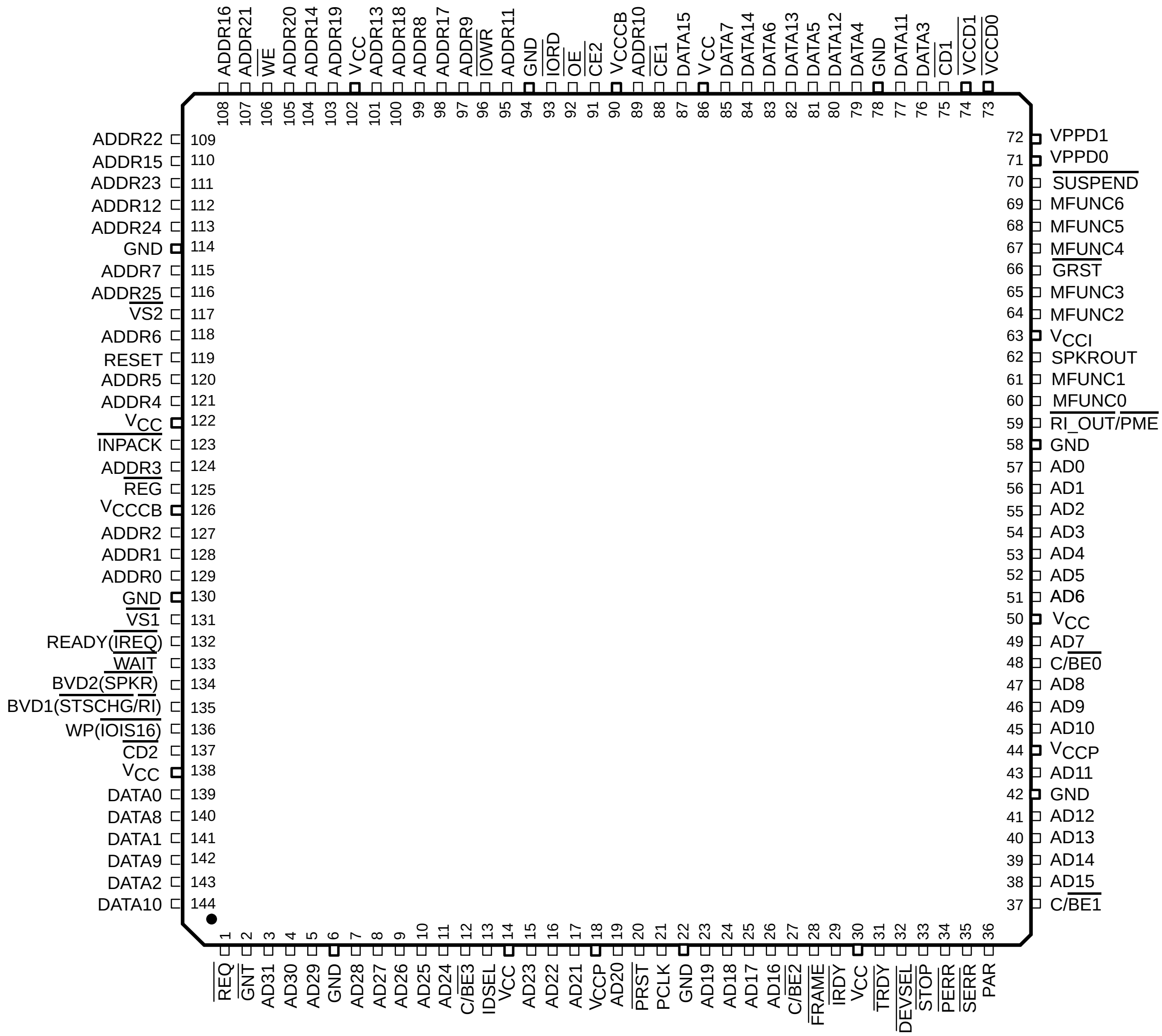


Figure 2–2. PCI-to-PC Card (16-Bit) Terminal Diagram

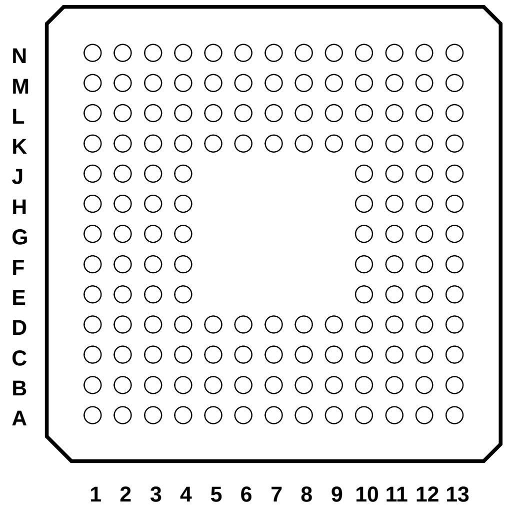


Figure 2–3. GGU Package Terminal Diagram

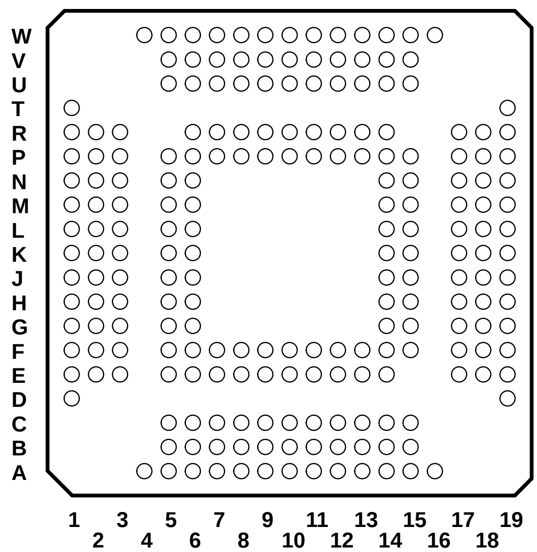


Figure 2–4. GHK Package Terminal Diagram

Table 2–1 shows the terminal assignments for the 144-terminal GGU CardBus and 16-bit PC Card signal names. Table 2–2 shows the terminal assignments for the 144-terminal PGE CardBus and 16-bit PC Card signal names. Table 2–3 shows the terminal assignments for the 209-terminal GHK CardBus and 16-bit PC Card signal names. Table 2–4 shows the CardBus PC Card signal names, sorted alphabetically to the GGU/PGE/GHK terminal numbers. Table 2–5 shows the 16-bit PC Card signal names, sorted alphabetically to the GGU/PGE/GHK terminal numbers.

Table 2–1. CardBus and 16-Bit PC Card Signal Names by GGU Terminal Number

TERM. NO.	SIGNAL NAME		TERM. NO.	SIGNAL NAME		TERM. NO.	SIGNAL NAME		TERM. NO.	SIGNAL NAME	
	CARD BUS	16-BIT		CARD BUS	16-BIT		CARD BUS	16-BIT		CARD BUS	16-BIT
A1	REQ	REQ	C11	CGNT	WE	G10	CAD11	OE	L4	GND	GND
A2	CRSVD	DATA2	C12	CSTOP	ADDR20	G11	CAD10	CE2	L5	AD9	AD9
A3	CAD28	DATA8	C13	CPERR	ADDR14	G12	CAD9	ADDR10	L6	V_{CC}	V_{CC}
A4	CCD2	CD2	D1	AD27	AD27	G13	V_{CCCB}	V_{CCCB}	L7	AD2	AD2
A5	CSERR	WAIT	D2	AD28	AD28	H1	PCLK	PCLK	L8	RI_OUT/ PME	RI_OUT/ PME
A6	CAD26	ADDR0	D3	GND	GND	H2	GND	GND	L9	V_{CCI}	V_{CCI}
A7	V_{CCCB}	V_{CCCB}	D4	AD29	AD29	H3	AD19	AD19	L10	MFUNC4	MFUNC4
A8	CAD23	ADDR3	D5	CCLKRUN	WP (IOIS16)	H4	AD18	AD18	L11	SUSPEND	SUSPEND
A9	CAD21	ADDR5	D6	CINT	READY (IREQ)	H10	CAD7	DATA7	L12	CCD1	CD1
A10	CAD19	ADDR25	D7	CAD25	ADDR1	H11	V_{CC}	V_{CC}	L13	CAD0	DATA3
A11	CC/BE2	ADDR12	D8	CAD22	ADDR4	H12	CAD8	DATA15	M1	SERR	SERR
A12	CIRDY	ADDR15	D9	CVS2	VS2	H13	CC/BE0	CE1	M2	PAR	PAR
A13	CTRDY	ADDR22	D10	CAD17	ADDR24	J1	AD17	AD17	M3	AD14	AD14
B1	GNT	GNT	D11	CBLOCK	ADDR19	J2	AD16	AD16	M4	AD11	AD11
B2	CAD31	DATA10	D12	V_{CC}	V_{CC}	J3	C/BE2	C/BE2	M5	AD8	AD8
B3	CAD29	DATA1	D13	CPAR	ADDR13	J4	FRAME	FRAME	M6	AD6	AD6
B4	V_{CC}	V_{CC}	E1	C/BE3	C/BE3	J10	CAD3	DATA5	M7	AD4	AD4
B5	CAUDIO	BVD2 (SPKR)	E2	AD24	AD24	J11	CAD6	DATA13	M8	GND	GND
B6	GND	GND	E3	AD25	AD25	J12	CAD5	DATA6	M9	SPKROUT	SPKROUT
B7	CC/BE3	REG	E4	AD26	AD26	J13	CRSVD	DATA14	M10	GRST	GRST
B8	CREQ	INPACK	E10	CRSVD	ADDR18	K1	IRDY	IRDY	M11	MFUNC6	MFUNC6
B9	CRST	RESET	E11	CC/BE1	ADDR8	K2	V_{CC}	V_{CC}	M12	VPPD1	VPPD1
B10	CAD18	ADDR7	E12	CAD16	ADDR17	K3	TRDY	TRDY	M13	VCCD1	VCCD1
B11	CFRAME	ADDR23	E13	CAD14	ADDR9	K4	AD12	AD12	N1	C/BE1	C/BE1
B12	CCLK	ADDR16	F1	AD22	AD22	K5	AD10	AD10	N2	AD15	AD15
B13	CDEVSEL	ADDR21	F2	AD23	AD23	K6	AD7	AD7	N3	AD13	AD13
C1	AD30	AD30	F3	V_{CC}	V_{CC}	K7	AD1	AD1	N4	V_{CCP}	V_{CCP}
C2	AD31	AD31	F4	IDSEL	IDSEL	K8	MFUNC0	MFUNC0	N5	C/BE0	C/BE0
C3	CAD30	DATA9	F10	CAD15	IOWR	K9	MFUNC2	MFUNC2	N6	AD5	AD5
C4	CAD27	DATA0	F11	CAD12	ADDR11	K10	CAD2	DATA11	N7	AD3	AD3
C5	CSTSCHG	BVD1 (STSCHG/RI)	F12	GND	GND	K11	GND	GND	N8	AD0	AD0
C6	CVS1	VS1	F13	CAD13	IORD	K12	CAD1	DATA4	N9	MFUNC1	MFUNC1
C7	CAD24	ADDR2	G1	V_{CCP}	V_{CCP}	K13	CAD4	DATA12	N10	MFUNC3	MFUNC3
C8	V_{CC}	V_{CC}	G2	AD21	AD21	L1	DEVSEL	DEVSEL	N11	MFUNC5	MFUNC5
C9	CAD20	ADDR6	G3	AD20	AD20	L2	STOP	STOP	N12	VPPD0	VPPD0
C10	GND	GND	G4	PRST	PRST	L3	PERR	PERR	N13	VCCD0	VCCD0

Table 2–2. CardBus and 16-Bit PC Card Signal Names by PGE Terminal Number

TERM NO.	SIGNAL NAME		TERM NO.	SIGNAL NAME		TERM NO.	SIGNAL NAME		TERM NO.	SIGNAL NAME	
	CARD BUS	16-BIT		CARD BUS	16-BIT		CARD BUS	16-BIT		CARD BUS	16-BIT
1	$\overline{\text{REQ}}$	$\overline{\text{REQ}}$	37	C/$\overline{\text{BE1}}$	C/$\overline{\text{BE1}}$	73	$\overline{\text{VCCD0}}$	$\overline{\text{VCCD0}}$	109	$\overline{\text{CTRDY}}$	ADDR22
2	$\overline{\text{GNT}}$	$\overline{\text{GNT}}$	38	AD15	AD15	74	$\overline{\text{VCCD1}}$	$\overline{\text{VCCD1}}$	110	$\overline{\text{CIRDY}}$	ADDR15
3	AD31	AD31	39	AD14	AD14	75	$\overline{\text{CCD1}}$	$\overline{\text{CD1}}$	111	$\overline{\text{CFRAME}}$	ADDR23
4	AD30	AD30	40	AD13	AD13	76	CAD0	DATA3	112	CC/$\overline{\text{BE2}}$	ADDR12
5	AD29	AD29	41	AD12	AD12	77	CAD2	DATA11	113	CAD17	ADDR24
6	GND	GND	42	GND	GND	78	GND	GND	114	GND	GND
7	AD28	AD28	43	AD11	AD11	79	CAD1	DATA4	115	CAD18	ADDR7
8	AD27	AD27	44	V_{CCP}	V_{CCP}	80	CAD4	DATA12	116	CAD19	ADDR25
9	AD26	AD26	45	AD10	AD10	81	CAD3	DATA5	117	CVS2	$\overline{\text{VS2}}$
10	AD25	AD25	46	AD9	AD9	82	CAD6	DATA13	118	CAD20	ADDR6
11	AD24	AD24	47	AD8	AD8	83	CAD5	DATA6	119	$\overline{\text{CRST}}$	RESET
12	C/$\overline{\text{BE3}}$	C/$\overline{\text{BE3}}$	48	C/$\overline{\text{BE0}}$	C/$\overline{\text{BE0}}$	84	CRSVD	DATA14	120	CAD21	ADDR5
13	IDSEL	IDSEL	49	AD7	AD7	85	CAD7	DATA7	121	CAD22	ADDR4
14	V_{CC}	V_{CC}	50	V_{CC}	V_{CC}	86	V_{CC}	V_{CC}	122	V_{CC}	V_{CC}
15	AD23	AD23	51	AD6	AD6	87	CAD8	DATA15	123	$\overline{\text{CREQ}}$	$\overline{\text{INPACK}}$
16	AD22	AD22	52	AD5	AD5	88	CC/$\overline{\text{BE0}}$	$\overline{\text{CE1}}$	124	CAD23	ADDR3
17	AD21	AD21	53	AD4	AD4	89	CAD9	ADDR10	125	CC/$\overline{\text{BE3}}$	$\overline{\text{REG}}$
18	V_{CCP}	V_{CCP}	54	AD3	AD3	90	V_{CCCB}	V_{CCCB}	126	V_{CCCB}	V_{CCCB}
19	AD20	AD20	55	AD2	AD2	91	CAD10	$\overline{\text{CE2}}$	127	CAD24	ADDR2
20	$\overline{\text{PRST}}$	$\overline{\text{PRST}}$	56	AD1	AD1	92	CAD11	$\overline{\text{OE}}$	128	CAD25	ADDR1
21	PCLK	PCLK	57	AD0	AD0	93	CAD13	$\overline{\text{IORD}}$	129	CAD26	ADDR0
22	GND	GND	58	GND	GND	94	GND	GND	130	GND	GND
23	AD19	AD19	59	$\overline{\text{RI_OUT}}$/ $\overline{\text{PME}}$	$\overline{\text{RI_OUT}}$/ $\overline{\text{PME}}$	95	CAD12	ADDR11	131	CVS1	$\overline{\text{VS1}}$
24	AD18	AD18	60	MFUNC0	MFUNC0	96	CAD15	$\overline{\text{IOWR}}$	132	$\overline{\text{CINT}}$	READY ($\overline{\text{IREQ}}$)
25	AD17	AD17	61	MFUNC1	MFUNC1	97	CAD14	ADDR9	133	$\overline{\text{CSERR}}$	$\overline{\text{WAIT}}$
26	AD16	AD16	62	SPKROUT	SPKROUT	98	CAD16	ADDR17	134	CAUDIO	BVD2 ($\overline{\text{SPKR}}$)
27	C/$\overline{\text{BE2}}$	C/$\overline{\text{BE2}}$	63	V_{CCI}	V_{CCI}	99	CC/$\overline{\text{BE1}}$	ADDR8	135	CSTSCHG	BVD1 ($\overline{\text{STSCHG}}$/$\overline{\text{RI}}$)
28	$\overline{\text{FRAME}}$	$\overline{\text{FRAME}}$	64	MFUNC2	MFUNC2	100	CRSVD	ADDR18	136	$\overline{\text{CCLKRUN}}$	WP ($\overline{\text{IOIS16}}$)
29	$\overline{\text{IRDY}}$	$\overline{\text{IRDY}}$	65	MFUNC3	MFUNC3	101	CPAR	ADDR13	137	$\overline{\text{CCD2}}$	$\overline{\text{CD2}}$
30	V_{CC}	V_{CC}	66	$\overline{\text{GRST}}$	$\overline{\text{GRST}}$	102	V_{CC}	V_{CC}	138	V_{CC}	V_{CC}
31	$\overline{\text{TRDY}}$	$\overline{\text{TRDY}}$	67	MFUNC4	MFUNC4	103	$\overline{\text{CBLOCK}}$	ADDR19	139	CAD27	DATA0
32	$\overline{\text{DEVSEL}}$	$\overline{\text{DEVSEL}}$	68	MFUNC5	MFUNC5	104	$\overline{\text{CPERR}}$	ADDR14	140	CAD28	DATA8
33	$\overline{\text{STOP}}$	$\overline{\text{STOP}}$	69	MFUNC6	MFUNC6	105	$\overline{\text{CSTOP}}$	ADDR20	141	CAD29	DATA1
34	$\overline{\text{PERR}}$	$\overline{\text{PERR}}$	70	$\overline{\text{SUSPEND}}$	$\overline{\text{SUSPEND}}$	106	$\overline{\text{CGNT}}$	$\overline{\text{WE}}$	142	CAD30	DATA9
35	$\overline{\text{SERR}}$	$\overline{\text{SERR}}$	71	VPPD0	VPPD0	107	$\overline{\text{CDEVSEL}}$	ADDR21	143	CRSVD	DATA2
36	PAR	PAR	72	VPPD1	VPPD1	108	CCLK	ADDR16	144	CAD31	DATA10

Table 2–3. CardBus and 16-Bit PC Card Signal Names by GHK Terminal Number

TERM NO.	SIGNAL NAME		TERM NO.	SIGNAL NAME		TERM NO.	SIGNAL NAME	
	CARDBUS	16-BIT		CARDBUS	16-BIT		CARDBUS	16-BIT
A4	NC	NC	E9	CAD29	DATA1	H17	CAD11	$\overline{OE}$
A5	NC	NC	E10	CSTSCHG	BVD1($\overline{STSCHG}$/$\overline{RI}$)	H18	CAD10	$\overline{CE2}$
A6	NC	NC	E11	GND	GND	H19	V_{CCCB}	V_{CCCB}
A7	NC	NC	E12	$\overline{CREQ}$	$\overline{INPACK}$	J1	AD31	AD31
A8	CAD30	DATA9	E13	CVS2	$\overline{VS2}$	J2	AD30	AD30
A9	$\overline{CCD2}$	$\overline{CD2}$	E14	$\overline{CFRAME}$	ADDR23	J3	AD29	AD29
A10	$\overline{CINT}$	READY($\overline{IREQ}$)	E17	$\overline{CDEVSEL}$	ADDR21	J5	GND	GND
A11	CAD24	ADDR2	E18	$\overline{CSTOP}$	ADDR20	J6	AD28	AD28
A12	V_{CCCB}	V_{CCCB}	E19	$\overline{CBLOCK}$	ADDR19	J14	CC/$\overline{BE0}$	$\overline{CE1}$
A13	V_{CC}	V_{CC}	F1	NC	NC	J15	CAD9	ADDR10
A14	CAD20	ADDR6	F2	NC	NC	J17	CAD8	DATA15
A15	GND	GND	F3	NC	NC	J18	V_{CC}	V_{CC}
A16	$\overline{CTRDY}$	ADDR22	F5	NC	NC	J19	CAD7	DATA7
B5	NC	NC	F6	NC	NC	K1	AD27	AD27
B6	NC	NC	F7	NC	NC	K2	AD26	AD26
B7	NC	NC	F8	NC	NC	K3	AD25	AD25
B8	CRSVD	DATA2	F9	CAD28	DATA8	K5	AD24	AD24
B9	V_{CC}	V_{CC}	F10	$\overline{CCLKRUN}$	WP($\overline{IOIS16}$)	K6	C/$\overline{BE3}$	C/$\overline{BE3}$
B10	$\overline{CSERR}$	$\overline{WAIT}$	F11	CVS1	$\overline{VS1}$	K14	CRSVD	DATA14
B11	CAD25	ADDR1	F12	$\overline{CRST}$	RESET	K15	CAD5	DATA6
B12	CC/$\overline{BE3}$	$\overline{REG}$	F13	CC/$\overline{BE2}$	ADDR12	K17	CAD6	DATA13
B13	CAD22	ADDR4	F14	$\overline{CPERR}$	ADDR14	K18	CAD3	DATA5
B14	CAD19	ADDR25	F15	$\overline{CGNT}$	$\overline{WE}$	K19	CAD4	DATA12
B15	CAD17	ADDR24	F17	V_{CC}	V_{CC}	L1	IDSEL	IDSEL
C5	NC	NC	F18	CRSVD	ADDR18	L2	V_{CC}	V_{CC}
C6	NC	NC	F19	CC/$\overline{BE1}$	ADDR8	L3	AD23	AD23
C7	NC	NC	G1	NC	NC	L5	AD21	AD21
C8	CAD31	DATA10	G2	NC	NC	L6	AD22	AD22
C9	CAD27	DATA0	G3	NC	NC	L14	CAD1	DATA4
C10	CAUDIO	BVD2($\overline{SPKR}$)	G5	NC	NC	L15	GND	GND
C11	CAD26	ADDR0	G6	NC	NC	L17	CAD2	DATA11
C12	CAD23	ADDR3	G14	CAD16	ADDR17	L18	CAD0	DATA3
C13	CAD21	ADDR5	G15	CPAR	ADDR13	L19	$\overline{CCD1}$	$\overline{CD1}$
C14	CAD18	ADDR7	G17	CAD14	ADDR9	M1	V_{CCP}	V_{CCP}
C15	$\overline{CIRDY}$	ADDR15	G18	CAD15	$\overline{IOWR}$	M2	AD20	AD20
D1	NC	NC	G19	CAD12	ADDR11	M3	$\overline{PRST}$	$\overline{PRST}$
D19	CCLK	ADDR16	H1	$\overline{GNT}$	$\overline{GNT}$	M5	GND	GND
E1	NC	NC	H2	$\overline{REQ}$	$\overline{REQ}$	M6	PCLK	PCLK
E2	NC	NC	H3	NC	NC	M14	NC	NC
E3	NC	NC	H5	NC	NC	M15	NC	NC
E6	NC	NC	H6	NC	NC	M17	NC	NC
E7	NC	NC	H14	CAD13	$\overline{IORD}$	M18	$\overline{VCCD0}$	$\overline{VCCD0}$
E8	NC	NC	H15	GND	GND	M19	$\overline{VCCD1}$	$\overline{VCCD1}$

Table 2–3. CardBus and 16-Bit PC Card Signal Names by GHK Terminal Number (Continued)

TERM. NO.	SIGNAL NAME		TERM. NO.	SIGNAL NAME		TERM. NO.	SIGNAL NAME	
	CARDBUS	16-BIT		CARDBUS	16-BIT		CARDBUS	16-BIT
N1	AD19	AD19	R1	$\overline{\text{TRDY}}$	$\overline{\text{TRDY}}$	U15	NC	NC
N2	AD18	AD18	R2	$\overline{\text{STOP}}$	$\overline{\text{STOP}}$	V5	AD12	AD12
N3	AD17	AD17	R3	$\overline{\text{SERR}}$	$\overline{\text{SERR}}$	V6	V_{CCP}	V_{CCP}
N5	$\overline{\text{IRDY}}$	$\overline{\text{IRDY}}$	R6	AD14	AD14	V7	AD7	AD7
N6	AD16	AD16	R7	AD10	AD10	V8	AD4	AD4
N14	NC	NC	R8	AD6	AD6	V9	AD1	AD1
N15	NC	NC	R9	GND	GND	V10	MFUNC1	MFUNC1
N17	NC	NC	R10	V_{CCI}	V_{CCI}	V11	$\overline{\text{GRST}}$	$\overline{\text{GRST}}$
N18	NC	NC	R11	MFUNC6	MFUNC6	V12	VPPD0	VPPD0
N19	NC	NC	R12	NC	NC	V13	NC	NC
P1	C/$\overline{\text{BE2}}$	C/$\overline{\text{BE2}}$	R13	NC	NC	V14	NC	NC
P2	$\overline{\text{FRAME}}$	$\overline{\text{FRAME}}$	R14	NC	NC	V15	NC	NC
P3	V_{CC}	V_{CC}	R17	NC	NC	W4	C/$\overline{\text{BE1}}$	C/$\overline{\text{BE1}}$
P5	$\overline{\text{PERR}}$	$\overline{\text{PERR}}$	R18	NC	NC	W5	GND	GND
P6	$\overline{\text{DEVSEL}}$	$\overline{\text{DEVSEL}}$	R19	NC	NC	W6	AD9	AD9
P7	AD13	AD13	T1	PAR	PAR	W7	V_{CC}	V_{CC}
P8	AD8	AD8	T19	NC	NC	W8	AD3	AD3
P9	$\overline{\text{RI_OUT}}$/$\overline{\text{PME}}$	$\overline{\text{RI_OUT}}$/$\overline{\text{PME}}$	U5	AD15	AD15	W9	AD2	AD2
P10	MFUNC2	MFUNC2	U6	AD11	AD11	W10	MFUNC0	MFUNC0
P11	MFUNC5	MFUNC5	U7	C/$\overline{\text{BE0}}$	C/$\overline{\text{BE0}}$	W11	MFUNC3	MFUNC3
P12	NC	NC	U8	AD5	AD5	W12	$\overline{\text{SUSPEND}}$	$\overline{\text{SUSPEND}}$
P13	NC	NC	U9	AD0	AD0	W13	NC	NC
P14	NC	NC	U10	SPKROUT	SPKROUT	W14	NC	NC
P15	NC	NC	U11	MFUNC4	MFUNC4	W15	NC	NC
P17	NC	NC	U12	VPPD1	VPPD1	W16	NC	NC
P18	NC	NC	U13	NC	NC			
P19	NC	NC	U14	NC	NC			

Table 2–4. CardBus PC Card Signal Names Sorted Alphabetically to GGU/PGE/GHK Terminal Number

SIGNAL NAME	TERM. NO.			SIGNAL NAME	TERM. NO.			SIGNAL NAME	TERM. NO.		
	GGU	PGE	GHK		GGU	PGE	GHK		GGU	PGE	GHK
AD0	N8	57	U9	CAD11	G10	92	H17	$\overline{\text{CRST}}$	B9	119	F12
AD1	K7	56	V9	CAD12	F11	95	G19	CRSVD	A2	143	B8
AD2	L7	55	W9	CAD13	F13	93	H14	CRSVD	E10	100	F18
AD3	N7	54	W8	CAD14	E13	97	G17	CRSVD	J13	84	K14
AD4	M7	53	V8	CAD15	F10	96	G18	$\overline{\text{CSERR}}$	A5	133	B10
AD5	N6	52	U8	CAD16	E12	98	G14	$\overline{\text{CSTOP}}$	C12	105	E18
AD6	M6	51	R8	CAD17	D10	113	B15	CSTSCHG	C5	135	E10
AD7	K6	49	V7	CAD18	B10	115	C14	$\overline{\text{CTRDY}}$	A13	109	A16
AD8	M5	47	P8	CAD19	A10	116	B14	CVS1	C6	131	F11
AD9	L5	46	W6	CAD20	C9	118	A14	CVS2	D9	117	E13
AD10	K5	45	R7	CAD21	A9	120	C13	$\overline{\text{DEVSEL}}$	L1	32	P6
AD11	M4	43	U6	CAD22	D8	121	B13	$\overline{\text{FRAME}}$	J4	28	P2
AD12	K4	41	V5	CAD23	A8	124	C12	GND	D3	6	A15
AD13	N3	40	P7	CAD24	C7	127	A11	GND	H2	22	E11
AD14	M3	39	R6	CAD25	D7	128	B11	GND	L4	42	H15
AD15	N2	38	U5	CAD26	A6	129	C11	GND	M8	58	J5
AD16	J2	26	N6	CAD27	C4	139	C9	GND	K11	78	L15
AD17	J1	25	N3	CAD28	A3	140	F9	GND	F12	94	M5
AD18	H4	24	N2	CAD29	B3	141	E9	GND	C10	114	R9
AD19	H3	23	N1	CAD30	C3	142	A8	GND	B6	130	W5
AD20	G3	19	M2	CAD31	B2	144	C8	$\overline{\text{GNT}}$	B1	2	H1
AD21	G2	17	L5	CAUDIO	B5	134	C10	$\overline{\text{GRST}}$	M10	66	V11
AD22	F1	16	L6	C/$\overline{\text{BE0}}$	N5	48	U7	IDSEL	F4	13	L1
AD23	F2	15	L3	C/$\overline{\text{BE1}}$	N1	37	W4	$\overline{\text{IRDY}}$	K1	29	N5
AD24	E2	11	K5	C/$\overline{\text{BE2}}$	J3	27	P1	MFUNC0	K8	60	W10
AD25	E3	10	K3	C/$\overline{\text{BE3}}$	E1	12	K6	MFUNC1	N9	61	V10
AD26	E4	9	K2	$\overline{\text{CBLOCK}}$	D11	103	E19	MFUNC2	K9	64	P10
AD27	D1	8	K1	CC/$\overline{\text{BE0}}$	H13	88	J14	MFUNC3	N10	65	W11
AD28	D2	7	J6	CC/$\overline{\text{BE1}}$	E11	99	F19	MFUNC4	L10	67	U11
AD29	D4	5	J3	CC/$\overline{\text{BE2}}$	A11	112	F13	MFUNC5	N11	68	P11
AD30	C1	4	J2	CC/$\overline{\text{BE3}}$	B7	125	B12	MFUNC6	M11	69	R11
AD31	C2	3	J1	$\overline{\text{CCD1}}$	L12	75	L19	PAR	M2	36	T1
CAD0	L13	76	L18	$\overline{\text{CCD2}}$	A4	137	A9	PCLK	H1	21	M6
CAD1	K12	79	L14	CCLK	B12	108	D19	$\overline{\text{PERR}}$	L3	34	P5
CAD2	K10	77	L17	$\overline{\text{CCLKRUN}}$	D5	136	F10	$\overline{\text{PRST}}$	G4	20	M3
CAD3	J10	81	K18	$\overline{\text{CDEVSEL}}$	B13	107	E17	$\overline{\text{REQ}}$	A1	1	H2
CAD4	K13	80	K19	$\overline{\text{CFRAME}}$	B11	111	E14	$\overline{\text{RI_OUT}}$/$\overline{\text{PME}}$	L8	59	P9
CAD5	J12	83	K15	$\overline{\text{CGNT}}$	C11	106	F15	$\overline{\text{SERR}}$	M1	35	R3
CAD6	J11	82	K17	$\overline{\text{CINT}}$	D6	132	A10	SPKROUT	M9	62	U10
CAD7	H10	85	J19	$\overline{\text{CIRDY}}$	A12	110	C15	$\overline{\text{STOP}}$	L2	33	R2
CAD8	H12	87	J17	CPAR	D13	101	G15	$\overline{\text{SUSPEND}}$	L11	70	W12
CAD9	G12	89	J15	$\overline{\text{CPERR}}$	C13	104	F14	$\overline{\text{TRDY}}$	K3	31	R1
CAD10	G11	91	H18	$\overline{\text{CREQ}}$	B8	123	E12	V_{CC}	F3	14	A13

Table 2–4. CardBus PC Card Signal Names Sorted Alphabetically to GGU/PGE/GHK Terminal Number (Continued)

SIGNAL NAME	TERM. NO.			SIGNAL NAME	TERM. NO.			SIGNAL NAME	TERM. NO.		
	GGU	PGE	GHK		GGU	PGE	GHK		GGU	PGE	GHK
V_{CC}	K2	30	B9	V_{CC}	B4	138	W7	V_{CCI}	L9	63	R10
V_{CC}	L6	50	F17	V_{CCCB}	G13	90	A12	V_{CCP}	G1	18	M1
V_{CC}	H11	86	J18	V_{CCCB}	A7	126	H19	V_{CCP}	N4	44	V6
V_{CC}	D12	102	L2	$\overline{\text{VCCD0}}$	N13	73	M18	VPPD0	N12	71	V12
V_{CC}	C8	122	P3	$\overline{\text{VCCD1}}$	M13	74	M19	VPPD1	M12	72	U12

Table 2–5. 16-Bit PC Card Signal Names Sorted Alphabetically to GGU/PGE/GHK Terminal Number

SIGNAL NAME	TERM. NO.			SIGNAL NAME	TERM. NO.			SIGNAL NAME	TERM. NO.		
	GGU	PGE	GHK		GGU	PGE	GHK		GGU	PGE	GHK
AD0	N8	57	U9	ADDR11	F11	95	G19	GND	D3	6	A15
AD1	K7	56	V9	ADDR12	A11	112	F13	GND	H2	22	E11
AD2	L7	55	W9	ADDR13	D13	101	G15	GND	L4	42	H15
AD3	N7	54	W8	ADDR14	C13	104	F14	GND	M8	58	J5
AD4	M7	53	V8	ADDR15	A12	110	C15	GND	K11	78	L15
AD5	N6	52	U8	ADDR16	B12	108	D19	GND	F12	94	M5
AD6	M6	51	R8	ADDR17	E12	98	G14	GND	C10	114	R9
AD7	K6	49	V7	ADDR18	E10	100	F18	GND	B6	130	W5
AD8	M5	47	P8	ADDR19	D11	103	E19	GNT	B1	2	H1
AD9	L5	46	W6	ADDR20	C12	105	E18	GRST	M10	66	V11
AD10	K5	45	R7	ADDR21	B13	107	E17	IDSEL	F4	13	L1
AD11	M4	43	U6	ADDR22	A13	109	A16	INPACK	B8	123	E12
AD12	K4	41	V5	ADDR23	B11	111	E14	IORD	F13	93	H14
AD13	N3	40	P7	ADDR24	D10	113	B15	IOWR	F10	96	G18
AD14	M3	39	R6	ADDR25	A10	116	B14	IRDY	K1	29	N5
AD15	N2	38	U5	BVD1(STSCHG/RI)	C5	135	E10	MFUNC0	K8	60	W10
AD16	J2	26	N6	BVD2(SPKR)	B5	134	C10	MFUNC1	N9	61	V10
AD17	J1	25	N3	C/BE0	N5	48	U7	MFUNC2	K9	64	P10
AD18	H4	24	N2	C/BE1	N1	37	W4	MFUNC3	N10	65	W11
AD19	H3	23	N1	C/BE2	J3	27	P1	MFUNC4	L10	67	U11
AD20	G3	19	M2	C/BE3	E1	12	K6	MFUNC5	N11	68	P11
AD21	G2	17	L5	CD1	L12	75	L19	MFUNC6	M11	69	R11
AD22	F1	16	L6	CD2	A4	137	A9	OE	G10	92	H17
AD23	F2	15	L3	CE1	H13	88	J14	PAR	M2	36	T1
AD24	E2	11	K5	CE2	G11	91	H18	PCLK	H1	21	M6
AD25	E3	10	K3	DATA0	C4	139	C9	PERR	L3	34	P5
AD26	E4	9	K2	DATA1	B3	141	E9	PRST	G4	20	M3
AD27	D1	8	K1	DATA2	A2	143	B8	READY(IREQ)	D6	132	A10
AD28	D2	7	J6	DATA3	L13	76	L18	REG	B7	125	B12
AD29	D4	5	J3	DATA4	K12	79	L14	REQ	A1	1	H2
AD30	C1	4	J2	DATA5	J10	81	K18	RESET	B9	119	F12
AD31	C2	3	J1	DATA6	J12	83	K15	RI_OUT/PME	L8	59	P9
ADDR0	A6	129	C11	DATA7	H10	85	J19	SERR	M1	35	R3
ADDR1	D7	128	B11	DATA8	A3	140	F9	SPKROUT	M9	62	U10
ADDR2	C7	127	A11	DATA9	C3	142	A8	STOP	L2	33	R2
ADDR3	A8	124	C12	DATA10	B2	144	C8	SUSPEND	L11	70	W12
ADDR4	D8	121	B13	DATA11	K10	77	L17	TRDY	K3	31	R1
ADDR5	A9	120	C13	DATA12	K13	80	K19	V_{CC}	F3	14	A13
ADDR6	C9	118	A14	DATA13	J11	82	K17	V_{CC}	K2	30	B9
ADDR7	B10	115	C14	DATA14	J13	84	K14	V_{CC}	L6	50	F17
ADDR8	E11	99	F19	DATA15	H12	87	J17	V_{CC}	H11	86	J18
ADDR9	E13	97	G17	DEVSEL	L1	32	P6	V_{CC}	D12	102	L2
ADDR10	G12	89	J15	FRAME	J4	28	P2	V_{CC}	C8	122	P3

Table 2–5. 16-Bit PC Card Signal Names Sorted Alphabetically to GGU/PGE/GHK Terminal Number (Continued)

SIGNAL NAME	TERM. NO.			SIGNAL NAME	TERM. NO.			SIGNAL NAME	TERM. NO.		
	GGU	PGE	GHK		GGU	PGE	GHK		GGU	PGE	GHK
V_{CC}	B4	138	W7	V_{CCI}	L9	63	R10	$\overline{VS1}$	C6	131	F11
V_{CCCB}	G13	90	A12	V_{CCP}	G1	18	M1	$\overline{VS2}$	D9	117	E13
V_{CCCB}	A7	126	H19	V_{CCP}	N4	44	V6	$\overline{WAIT}$	A5	133	B10
$\overline{VCCD0}$	N13	73	M18	VPPD0	N12	71	V12	$\overline{WE}$	C11	106	F15
$\overline{VCCD1}$	M13	74	M19	VPPD1	M12	72	U12	WP($\overline{IOIS16}$)	D5	136	F10

The terminals are grouped in tables by functionality, such as PCI system function and power-supply function (see Table 2–6 through Table 2–16). The terminal numbers also are listed for convenient reference.

Table 2–6. Power-Supply Terminals

TERMINAL				DESCRIPTION
NAME	NUMBER			
	GGU	PGE	GHK	
GND	B6, C10, D3, F12, H2, K11, L4, M8	6, 22, 42, 58, 78, 94, 114, 130	A15, E11, H15, J5, L15, M5, R9, W5	Device ground terminals
V_{CC}	B4, C8, D12, F3, H11, K2, L6	14, 30, 50, 86, 102, 122, 138	A13, B9, F17, J18, L2, P3, W7	Power-supply terminal for core logic (3.3 V)
V_{CCCB}	A7, G13	90, 126	A12, H19	Clamp voltage for PC Card interface. Matches card signaling environment, 5 V or 3.3 V.
V_{CCI}	L9	63	R10	Clamp voltage for interrupt subsystem interface and miscellaneous I/O, 5 V or 3.3 V
V_{CCP}	G1, N4	18, 44	M1, V6	Clamp voltage for PCI signaling, 5 V or 3.3 V

Table 2–7. PC Card Power-Switch Terminals

TERMINAL				I/O	DESCRIPTION
NAME	NUMBER				
	GGU	PGE	GHK		
$\overline{\text{VCCD0}}$ $\overline{\text{VCCD1}}$	N13 M13	73 74	M18 M19	O	Logic controls to the TPS2211 PC Card power-switch interface to control AVCC.
VPPD0 VPPD1	N12 M12	71 72	V12 U12	O	Logic controls to the TPS2211 PC Card power-switch interface to control AVPP.

Table 2–8. PCI System Terminals

TERMINAL				I/O	DESCRIPTION
NAME	NUMBER				
	GGU	PGE	GHK		
$\overline{\text{GRST}}$	M10	66	V11	I	Global reset. When global reset is asserted, $\overline{\text{GRST}}$ causes the PCI1410A device to place all output buffers in a high-impedance state and reset all internal registers. When $\overline{\text{GRST}}$ is asserted, the device is completely in its default state. For systems that require wake-up from D3, $\overline{\text{GRST}}$ normally is asserted only during initial boot. $\overline{\text{PRST}}$ should be asserted following initial boot so that PME context is retained when transitioning from D3 to D0. For systems that do not require wake-up from D3, $\overline{\text{GRST}}$ should be tied to $\overline{\text{PRST}}$. When the $\overline{\text{SUSPEND}}$ mode is enabled, the device is protected from $\overline{\text{GRST}}$, and the internal registers are preserved. All outputs are placed in a high-impedance state.
PCLK	H1	21	M6	I	PCI bus clock. PCLK provides timing for all transactions on the PCI bus. All PCI signals are sampled at the rising edge of PCLK.
$\overline{\text{PRST}}$	G4	20	M3	I	PCI bus reset. When the PCI bus reset is asserted, $\overline{\text{PRST}}$ causes the PCI1410A device to place all output buffers in a high-impedance state and reset internal registers. When $\overline{\text{PRST}}$ is asserted, the device is completely nonfunctional. After $\overline{\text{PRST}}$ is deasserted, the PCI1410A device is in a default state. When the $\overline{\text{SUSPEND}}$ mode is enabled, the device is protected from $\overline{\text{PRST}}$, and the internal registers are preserved. All outputs are placed in a high-impedance state.

Table 2–9. PCI Address and Data Terminals

TERMINAL				I/O	DESCRIPTION
NAME	NUMBER				
	GGU	PGE	GHK		
AD31	C2	3	J1	I/O	PCI address/data bus. These signals make up the multiplexed PCI address and data bus on the primary interface. During the address phase of a primary bus PCI cycle, AD31–AD0 contain a 32-bit address or other destination information. During the data phase, AD31–AD0 contain data.
AD30	C1	4	J2		
AD29	D4	5	J3		
AD28	D2	7	J6		
AD27	D1	8	K1		
AD26	E4	9	K2		
AD25	E3	10	K3		
AD24	E2	11	K5		
AD23	F2	15	L3		
AD22	F1	16	L6		
AD21	G2	17	L5		
AD20	G3	19	M2		
AD19	H3	23	N1		
AD18	H4	24	N2		
AD17	J1	25	N3		
AD16	J2	26	N6		
AD15	N2	38	U5		
AD14	M3	39	R6		
AD13	N3	40	P7		
AD12	K4	41	V5		
AD11	M4	43	U6		
AD10	K5	45	R7		
AD9	L5	46	W6		
AD8	M5	47	P8		
AD7	K6	49	V7		
AD6	M6	51	R8		
AD5	N6	52	U8		
AD4	M7	53	V8		
AD3	N7	54	W8		
AD2	L7	55	W9		
AD1	K7	56	V9		
AD0	N8	57	U9		
C/BE3 C/BE2 C/BE1 C/BE0	E1 J3 N1 N5	12 27 37 48	K6 P1 W4 U7	I/O	PCI bus commands and byte enables. These signals are multiplexed on the same PCI terminals. During the address phase of a primary bus PCI cycle, C/BE3–C/BE0 define the bus command. During the data phase, this 4-bit bus is used as byte enables. The byte enables determine which byte paths of the full 32-bit data bus carry meaningful data. C/BE0 applies to byte 0 (AD7–AD0), C/BE1 applies to byte 1 (AD15–AD8), C/BE2 applies to byte 2 (AD23–AD16), and C/BE3 applies to byte 3 (AD31–AD24).
PAR	M2	36	T1	I/O	PCI bus parity. In all PCI bus read and write cycles, the PCI1410A device calculates even parity across the AD31–AD0 and C/BE3–C/BE0 buses. As an initiator during PCI cycles, the PCI1410A device outputs this parity indicator with a one-PCLK delay. As a target during PCI cycles, the calculated parity is compared to the initiator parity indicator. A compare error results in the assertion of a parity error (PERR).

Table 2–10. PCI Interface Control Terminals

TERMINAL				I/O	DESCRIPTION
NAME	NUMBER				
	GGU	PGE	GHK		
DEVSEL	L1	32	P6	I/O	PCI device select. The PCI1410A device asserts DEVSEL to claim a PCI cycle as the target device. As a PCI initiator on the bus, the PCI1410A device monitors DEVSEL until a target responds. If no target responds before timeout occurs, the PCI1410A device terminates the cycle with an initiator abort.
FRAME	J4	28	P2	I/O	PCI cycle frame. FRAME is driven by the initiator of a bus cycle. FRAME is asserted to indicate that a bus transaction is beginning, and data transfers continue while this signal is asserted. When FRAME is deasserted, the PCI bus transaction is in the final data phase.
GNT	B1	2	H1	I	PCI bus grant. GNT is driven by the PCI bus arbiter to grant the PCI1410A device access to the PCI bus after the current data transaction has completed. GNT may or may not follow a PCI bus request, depending on the PCI bus parking algorithm.
IDSEL	F4	13	L1	I	Initialization device select. IDSEL selects the PCI1410A device during configuration space accesses. IDSEL can be connected to one of the upper 24 PCI address lines on the PCI bus.
IRDY	K1	29	N5	I/O	PCI initiator ready. IRDY indicates the PCI bus initiator's ability to complete the current data phase of the transaction. A data phase is completed on a rising edge of PCLK, when both IRDY and TRDY are asserted. Until IRDY and TRDY are both sampled asserted, wait states are inserted.
PERR	L3	34	P5	I/O	PCI parity error indicator. PERR is driven by a PCI device to indicate that calculated parity does not match PAR when PERR is enabled through bit 6 (PERR_EN) of the command register (PCI offset 04h, see Section 4.4).
REQ	A1	1	H2	O	PCI bus request. REQ is asserted by the PCI1410A device to request access to the PCI bus as an initiator.
SERR	M1	35	R3	O	PCI system error. SERR is an output that is pulsed from the PCI1410A device when enabled through bit 8 (SERR_EN) of the command register (PCI offset 04h, see Section 4.4) indicating a system error has occurred. The PCI1410A device need not be the target of the PCI cycle to assert this signal. When SERR is enabled in the command register, this signal also pulses, indicating that an address parity error has occurred on a CardBus interface.
STOP	L2	33	R2	I/O	PCI cycle stop signal. STOP is driven by a PCI target to request the initiator to stop the current PCI bus transaction. STOP is used for target disconnects and is commonly asserted by target devices that do not support burst data transfers.
TRDY	K3	31	R1	I/O	PCI target ready. TRDY indicates the primary bus target's ability to complete the current data phase of the transaction. A data phase is completed on a rising edge of PCLK, when both IRDY and TRDY are asserted. Until both IRDY and TRDY are asserted, wait states are inserted.

Table 2–11. Multifunction and Miscellaneous Terminals

TERMINAL				I/O	DESCRIPTION
NAME	NUMBER				
	GGU	PGE	GHK		
MFUNC0	K8	60	W10	I/O	Multifunction terminal 0. MFUNC0 can be configured as parallel PCI interrupt INTA, GPI0, GPO0, socket activity LED output, zoomed-video (ZV) switching outputs, CardBus audio PWM, GPE, or a parallel IRQ. See Section 4.30, *Multifunction Routing Register*, for configuration details.
MFUNC1	N9	61	V10	I/O	Multifunction terminal 1. MFUNC1 can be configured as GPI1, GPO1, socket activity LED output, ZV switching outputs, CardBus audio PWM, GPE, or a parallel IRQ. See Section 4.30, *Multifunction Routing Register*, for configuration details. Serial data (SDA). When VCCD0 and VCCD1 are high after a PCI reset, the MFUNC1 terminal provides the SDA signaling for the serial bus interface. The two-terminal serial interface loads the subsystem identification and other register defaults from an EEPROM after a PCI reset. See Section 3.6.1, *Serial Bus Interface Implementation*, for details on other serial bus applications.
MFUNC2	K9	64	P10	I/O	Multifunction terminal 2. MFUNC2 can be configured as PC/PCI DMA request, GPI2, GPO2, socket activity LED output, ZV switching outputs, CardBus audio PWM, GPE, RI_OUT, or a parallel IRQ. See Section 4.30, *Multifunction Routing Register*, for configuration details.
MFUNC3	N10	65	W11	I/O	Multifunction terminal 3. MFUNC3 can be configured as a parallel IRQ or the serialized interrupt signal IRQSER. See Section 4.30, *Multifunction Routing Register*, for configuration details.
MFUNC4	L10	67	U11	I/O	Multifunction terminal 4. MFUNC4 can be configured as PCI LOCK, GPI3, GPO3, socket activity LED output, ZV switching outputs, CardBus audio PWM, GPE, RI_OUT, or a parallel IRQ. See Section 4.30, *Multifunction Routing Register*, for configuration details. Serial clock (SCL). When VCCD0 and VCCD1 are high after a PCI reset, the MFUNC4 terminal provides the SCL signaling for the serial bus interface. The two-terminal serial interface loads the subsystem identification and other register defaults from an EEPROM after a PCI reset. See Section 3.6.1, *Serial Bus Interface Implementation*, for details on other serial bus applications.
MFUNC5	N11	68	P11	I/O	Multifunction terminal 5. MFUNC5 can be configured as PC/PCI DMA grant, GPI4, GPO4, socket activity LED output, ZV switching outputs, CardBus audio PWM, GPE, or a parallel IRQ. See Section 4.30, *Multifunction Routing Register*, for configuration details.
MFUNC6	M11	69	R11	I/O	Multifunction terminal 6. MFUNC6 can be configured as a PCI CLKRUN or a parallel IRQ. See Section 4.30, *Multifunction Routing Register*, for configuration details.
RI_OUT/PME	L8	59	P9	O	Ring indicate out and power-management event output. Terminal provides an output for ring-indicate or PME signals.
SPKROUT	M9	62	U10	O	Speaker output. SPKROUT is the output to the host system that can carry SPKR or CAUDIO through the PCI1410A device from the PC Card interface. SPKROUT is driven as the exclusive-OR combination of card SPKR//CAUDIO inputs.
SUSPEND	L11	70	W12	I	Suspend. SUSPEND protects the internal registers from clearing when the GRST or PRST signal is asserted. See Section 3.8.4, *Suspend Mode*, for details.

Table 2–12. 16-Bit PC Card Address and Data Terminals

TERMINAL				I/O	DESCRIPTION
NAME	NUMBER				
	GGU	PGE	GHK		
ADDR25	A10	116	B14	O	PC Card address. 16-bit PC Card address lines. ADDR25 is the most significant bit.
ADDR24	D10	113	B15		
ADDR23	B11	111	E14		
ADDR22	A13	109	A16		
ADDR21	B13	107	E17		
ADDR20	C12	105	E18		
ADDR19	D11	103	E19		
ADDR18	E10	100	F18		
ADDR17	E12	98	G14		
ADDR16	B12	108	D19		
ADDR15	A12	110	C15		
ADDR14	C13	104	F14		
ADDR13	D13	101	G15		
ADDR12	A11	112	F13		
ADDR11	F11	95	G19		
ADDR10	G12	89	J15		
ADDR9	E13	97	G17		
ADDR8	E11	99	F19		
ADDR7	B10	115	C14		
ADDR6	C9	118	A14		
ADDR5	A9	120	C13		
ADDR4	D8	121	B13		
ADDR3	A8	124	C12		
ADDR2	C7	127	A11		
ADDR1	D7	128	B11		
ADDR0	A6	129	C11		
DATA15	H12	87	J17	I/O	PC Card data. 16-bit PC Card data lines. DATA15 is the most significant bit.
DATA14	J13	84	K14		
DATA13	J11	82	K17		
DATA12	K13	80	K19		
DATA11	K10	77	L17		
DATA10	B2	144	C8		
DATA9	C3	142	A8		
DATA8	A3	140	F9		
DATA7	H10	85	J19		
DATA6	J12	83	K15		
DATA5	J10	81	K18		
DATA4	K12	79	L14		
DATA3	L13	76	L18		
DATA2	A2	143	B8		
DATA1	B3	141	E9		
DATA0	C4	139	C9		

Table 2–13. 16-Bit PC Card Interface Control Terminals

TERMINAL				I/O	DESCRIPTION
NAME	NUMBER				
	GGU	PGE	GHK		
BVD1 (STSCHG/RI)	C5	135	E10	I	Battery voltage detect 1. BVD1 is generated by 16-bit memory PC Cards that include batteries. BVD1 is used with BVD2 as an indication of the condition of the batteries on a memory PC Card. Both BVD1 and BVD2 are high when the battery is good. When BVD2 is low and BVD1 is high, the battery is weak and should be replaced. When BVD1 is low, the battery is no longer serviceable and the data in the memory PC Card is lost. See Section 5.6, *ExCA Card Status-Change-Interrupt Configuration Register*, for enable bits. See Section 5.5, *ExCA Card Status-Change Register*, and Section 5.2, *ExCA Interface Status Register*, for the status bits for this signal. Status change. STSCHG is used to alert the system to a change in the READY, write protect, or battery voltage dead condition of a 16-bit I/O PC Card. Ring indicate. RI is used by 16-bit modem cards to indicate a ring detection.
BVD2 (SPKR)	B5	134	C10	I	Battery voltage detect 2. BVD2 is generated by 16-bit memory PC Cards that include batteries. BVD2 is used with BVD1 as an indication of the condition of the batteries on a memory PC Card. Both BVD1 and BVD2 are high when the battery is good. When BVD2 is low and BVD1 is high, the battery is weak and should be replaced. When BVD1 is low, the battery is no longer serviceable and the data in the memory PC Card is lost. See Section 5.6, *ExCA Card Status-Change-Interrupt Configuration Register*, for enable bits. See Section 5.5, *ExCA Card Status-Change Register*, and Section 5.2, *ExCA Interface Status Register*, for the status bits for this signal. Speaker. SPKR is an optional binary audio signal available only when the card and socket have been configured for the 16-bit I/O interface. The audio signals from cards A and B are combined by the PCI1410A device and are output on SPKROUT. DMA request. BVD2 can be used as the DMA request signal during DMA operations to a 16-bit PC Card that supports DMA. The PC Card asserts BVD2 to indicate a request for a DMA operation.
CD1 CD2	L12 A4	75 137	L19 A9	I	Card detect 1 and card detect 2. CD1 and CD2 are connected internally to ground on the PC Card. When a PC Card is inserted into a socket, CD1 and CD2 are pulled low. For signal status, see Section 5.2, *ExCA Interface Status Register*.
CE1 CE2	H13 G11	88 91	J14 H18	O	Card enable 1 and card enable 2. CE1 and CE2 enable even- and odd-numbered address bytes. CE1 enables even-numbered address bytes, and CE2 enables odd-numbered address bytes.
INPACK	B8	123	E12	I	Input acknowledge. INPACK is asserted by the PC Card when it can respond to an I/O read cycle at the current address. DMA request. INPACK can be used as the DMA request signal during DMA operations from a 16-bit PC Card that supports DMA. If it is used as a strobe, the PC Card asserts this signal to indicate a request for a DMA operation.
IORD	F13	93	H14	O	I/O read. IORD is asserted by the PCI1410A device to enable 16-bit I/O PC Card data output during host I/O read cycles. DMA write. IORD is used as the DMA write strobe during DMA operations from a 16-bit PC Card that supports DMA. The PCI1410A device asserts IORD during DMA transfers from the PC Card to host memory.
IOWR	F10	96	G18	O	I/O write. IOWR is driven low by the PCI1410A device to strobe write data into 16-bit I/O PC Cards during host I/O write cycles. DMA read. IOWR is used as the DMA write strobe during DMA operations from a 16-bit PC Card that supports DMA. The PCI1410A device asserts IOWR during transfers from host memory to the PC Card.
OE	G10	92	H17	O	Output enable. OE is driven low by the PCI1410A device to enable 16-bit memory PC Card data output during host memory read cycles. DMA terminal count. OE is used as terminal count (TC) during DMA operations to a 16-bit PC Card that supports DMA. The PCI1410A device asserts OE to indicate TC for a DMA write operation.

Table 2–13. 16-Bit PC Card Interface Control Terminals (Continued)

TERMINAL				I/O	DESCRIPTION
NAME	NUMBER				
	GGU	PGE	GHK		
READY (IREQ)	D6	132	A10	I	Ready. The ready function is provided by READY when the 16-bit PC Card and the host socket are configured for the memory-only interface. READY is driven low by the 16-bit memory PC Cards to indicate that the memory card circuits are busy processing a previous write command. READY is driven high when the 16-bit memory PC Card is ready to accept a new data-transfer command. Interrupt request. IREQ is asserted by a 16-bit I/O PC Card to indicate to the host that a device on the 16-bit I/O PC Card requires service by the host software. IREQ is high (deasserted) when no interrupt is requested.
REG	B7	125	B12	O	Attribute memory select. REG remains high for all common memory accesses. When REG is asserted, access is limited to attribute memory (OE or WE active) and to the I/O space (IORD or IOWR active). Attribute memory is a separately accessed section of card memory and generally is used to record card capacity and other configuration and attribute information. DMA acknowledge. REG is used as a DMA acknowledge (DACK) during DMA operations to a 16-bit PC Card that supports DMA. The PCI1410A device asserts REG to indicate a DMA operation. REG is used in conjunction with the DMA read (IOWR) or DMA write (IORD) strobes to transfer data.
RESET	B9	119	F12	O	PC Card reset. RESET forces a hard reset to a 16-bit PC Card.
VS1 VS2	C6 D9	131 117	F11 E13	I/O	Voltage sense 1 and voltage sense 2. VS1 and VS2, when used in conjunction with each other, determine the operating voltage of the PC Card.
WAIT	A5	133	B10	I	Bus cycle wait. WAIT is driven by a 16-bit PC Card to extend the completion of the memory or I/O cycle in progress.
WE	C11	106	F15	O	Write enable. WE is used to strobe memory write data into 16-bit memory PC Cards. WE also is used for memory PC Cards that employ programmable memory technologies. DMA terminal count. WE is used as TC during DMA operations to a 16-bit PC Card that supports DMA. The PCI1410A device asserts WE to indicate TC for a DMA read operation.
WP (IOIS16)	D5	136	F10	I	Write protect. WP applies to 16-bit memory PC Cards. WP reflects the status of the write-protect switch on 16-bit memory PC Cards. For 16-bit I/O cards, WP is used for the 16-bit port (IOIS16) function. I/O is 16 bits. IOIS16 applies to 16-bit I/O PC Cards. IOIS16 is asserted by the 16-bit PC Card when the address on the bus corresponds to an address to which the 16-bit PC Card responds, and the I/O port that is addressed is capable of 16-bit accesses. DMA request. WP can be used as the DMA request signal during DMA operations to a 16-bit PC Card that supports DMA. If used, the PC Card asserts WP to indicate a request for a DMA operation.

Table 2–14. CardBus PC Card Interface System Terminals

TERMINAL				I/O	DESCRIPTION
NAME	NUMBER				
	GGU	PGE	GHK		
CCLK	B12	108	D19	O	CardBus clock. CCLK provides synchronous timing for all transactions on the CardBus interface. All signals except CRST, CCLKRUN, CINT, CSTSCHG, CAUDIO, CCD2, CCD1, CVS2, and CVS1 are sampled on the rising edge of CCLK, and all timing parameters are defined with the rising edge of this signal. CCLK operates at the PCI bus clock frequency, but it can be stopped in the low state or slowed down for power savings.
CCLKRUN	D5	136	F10	I/O	CardBus clock run. CCLKRUN is used by a CardBus PC Card to request an increase in the CCLK frequency, and by the PCI1410A device to indicate that the CCLK frequency is going to be decreased.
CRST	B9	119	F12	O	CardBus reset. CRST brings CardBus PC Card-specific registers, sequencers, and signals to a known state. When CRST is asserted, all CardBus PC Card signals are placed in a high-impedance state, and the PCI1410A device drives these signals to a valid logic level. Assertion can be asynchronous to CCLK, but deassertion must be synchronous to CCLK.

Table 2–15. CardBus PC Card Address and Data Terminals

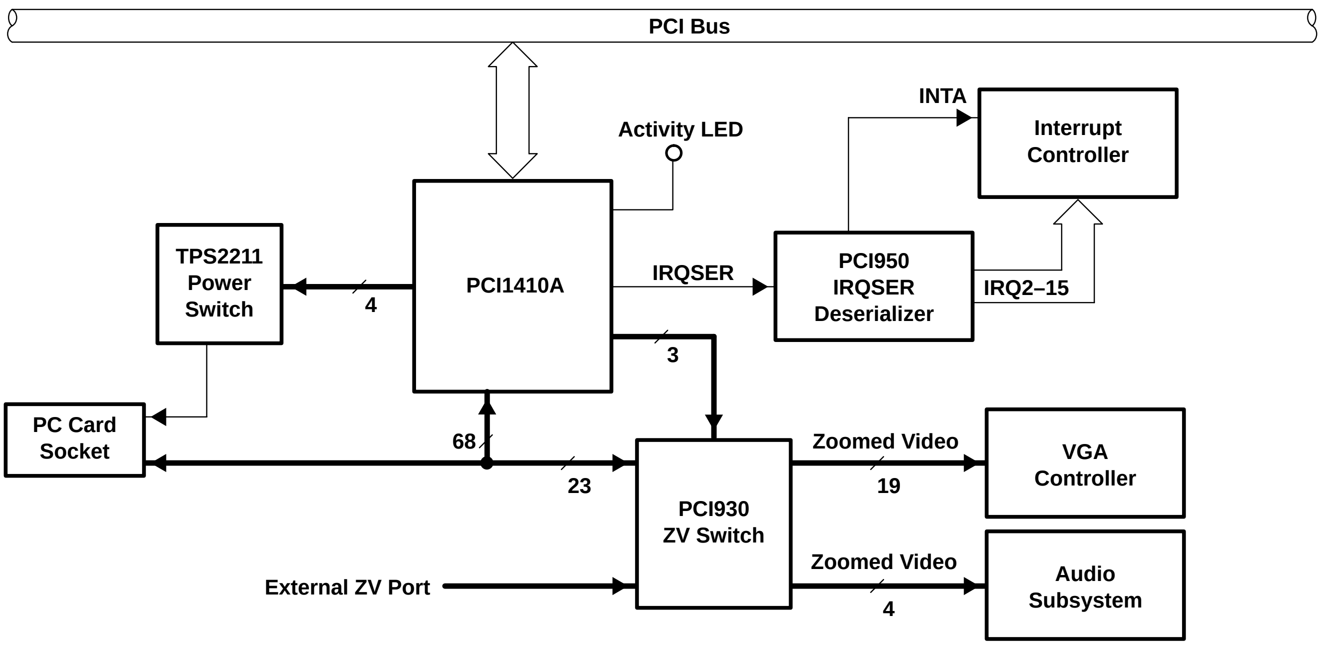
TERMINAL				I/O	DESCRIPTION
NAME	NUMBER				
	GGU	PGE	GHK		
CAD31	B2	144	C8	I/O	CardBus address and data. These signals make up the multiplexed CardBus address and data bus on the CardBus interface. During the address phase of a CardBus cycle, CAD31–CAD0 contain a 32-bit address. During the data phase of a CardBus cycle, CAD31–CAD0 contain data. CAD31 is the most significant bit.
CAD30	C3	142	A8		
CAD29	B3	141	E9		
CAD28	A3	140	F9		
CAD27	C4	139	C9		
CAD26	A6	129	C11		
CAD25	D7	128	B11		
CAD24	C7	127	A11		
CAD23	A8	124	C12		
CAD22	D8	121	B13		
CAD21	A9	120	C13		
CAD20	C9	118	A14		
CAD19	A10	116	B14		
CAD18	B10	115	C14		
CAD17	D10	113	B15		
CAD16	E12	98	G14		
CAD15	F10	96	G18		
CAD14	E13	97	G17		
CAD13	F13	93	H14		
CAD12	F11	95	G19		
CAD11	G10	92	H17		
CAD10	G11	91	H18		
CAD9	G12	89	J15		
CAD8	H12	87	J17		
CAD7	H10	85	J19		
CAD6	J11	82	K17		
CAD5	J12	83	K15		
CAD4	K13	80	K19		
CAD3	J10	81	K18		
CAD2	K10	77	L17		
CAD1	K12	79	L14		
CAD0	L13	76	L18		
CC/$\overline{\text{BE3}}$	B7	125	B12	I/O	CardBus bus commands and byte enables. CC/$\overline{\text{BE3}}$–CC/$\overline{\text{BE0}}$ are multiplexed on the same CardBus terminals. During the address phase of a CardBus cycle, CC/$\overline{\text{BE3}}$–CC/$\overline{\text{BE0}}$ define the bus command. During the data phase, this 4-bit bus is used as byte enables. The byte enables determine which byte paths of the full 32-bit data bus carry meaningful data. CC/$\overline{\text{BE0}}$ applies to byte 0 (CAD7–CAD0), CC/$\overline{\text{BE1}}$ applies to byte 1 (CAD15–CAD8), CC/$\overline{\text{BE2}}$ applies to byte 2 (CAD23–CAD16), and CC/$\overline{\text{BE3}}$ applies to byte 3 (CAD31–CAD24).
CC/$\overline{\text{BE2}}$	A11	112	F13		
CC/$\overline{\text{BE1}}$	E11	99	F19		
CC/$\overline{\text{BE0}}$	H13	88	J14		
CPAR	D13	101	G15	I/O	CardBus parity. In all CardBus read and write cycles, the PCI1410A device calculates even parity across the CAD and CC/$\overline{\text{BE}}$ buses. As an initiator during CardBus cycles, the PCI1410A device outputs CPAR with a one-CCLK delay. As a target during CardBus cycles, the calculated parity is compared to the initiator's parity indicator; a compare error results in a parity-error assertion.

Table 2–16. CardBus PC Card Interface Control Terminals

TERMINAL				I/O	DESCRIPTION
NAME	NUMBER				
	GGU	PGE	GHK		
CAUDIO	B5	134	C10	I	CardBus audio. CAUDIO is a digital input signal from a PC Card to the system speaker. The PCI1410A device supports the binary audio mode and outputs a binary signal from the card to SPKROUT.
CBLOCK	D11	103	E19	I/O	CardBus lock. CBLOCK is used to gain exclusive access to a target.
CCD1 CCD2	L12 A4	75 137	L19 A9	I	CardBus detect 1 and CardBus detect 2. CCD1 and CCD2 are used in conjunction with CVS1 and CVS2 to identify card insertion and interrogate cards to determine the operating voltage and card type.
CDEVSEL	B13	107	E17	I/O	CardBus device select. The PCI1410A device asserts CDEVSEL to claim a CardBus cycle as the target device. As a CardBus initiator on the bus, the PCI1410A device monitors CDEVSEL until a target responds. If no target responds before timeout occurs, the PCI1410A device terminates the cycle with an initiator abort.
CFRAME	B11	111	E14	I/O	CardBus cycle frame. CFRAME is driven by the initiator of a CardBus bus cycle. CFRAME is asserted to indicate that a bus transaction is beginning, and data transfers continue while this signal is asserted. When CFRAME is deasserted, the CardBus bus transaction is in the final data phase.
CGNT	C11	106	F15	O	CardBus bus grant. CGNT is driven by the PCI1410A device to grant a CardBus PC Card access to the CardBus bus after the current data transaction has been completed.
CINT	D6	132	A10	I	CardBus interrupt. CINT is asserted low by a CardBus PC Card to request interrupt servicing from the host.
CIRDY	A12	110	C15	I/O	CardBus initiator ready. CIRDY indicates the ability of the CardBus initiator to complete the current data phase of the transaction. A data phase is completed on a rising edge of CCLK when both CIRDY and CTRDY are asserted. Until both CIRDY and CTRDY are sampled asserted, wait states are inserted.
CPERR	C13	104	F14	I/O	CardBus parity error. CPERR reports parity errors during CardBus transactions, except during special cycles. It is driven low by a target two clocks following that data when a parity error is detected.
CREQ	B8	123	E12	I	CardBus request. CREQ indicates to the arbiter that the CardBus PC Card desires use of the CardBus bus as an initiator.
CSERR	A5	133	B10	I	CardBus system error. CSERR reports address parity errors and other system errors that could lead to catastrophic results. CSERR is driven by the card synchronous to CCLK. The PCI1410A device can report CSERR to the system by assertion of SERR on the PCI interface.
CSTOP	C12	105	E18	I/O	CardBus stop. CSTOP is driven by a CardBus target to request the initiator to stop the current CardBus transaction. CSTOP is used for target disconnects, and is commonly asserted by target devices that do not support burst data transfers.
CSTSCHG	C5	135	E10	I	CardBus status change. CSTSCHG alerts the system to a change in the card status, and is used as a wake-up mechanism.
CTRDY	A13	109	A16	I/O	CardBus target ready. CTRDY indicates the ability of the CardBus target ability to complete the current data phase of the transaction. A data phase is completed on a rising edge of CCLK, when both CIRDY and CTRDY are asserted; until this time, wait states are inserted.
CVS1 CVS2	C6 D9	131 117	F11 E13	I/O	CardBus voltage sense 1 and CardBus voltage sense 2. CVS1 and CVS2 are used in conjunction with CCD1 and CCD2 to identify card insertion and interrogate cards to determine the operating voltage and card type.

3 Feature/Protocol Descriptions

The following sections give an overview of the PCI1410A device. Figure 3–1 shows a simplified block diagram of the PCI1410A device. The PCI interface includes all address/data and control signals for PCI protocol. The interrupt interface includes terminals for parallel PCI, parallel ISA, and serialized PCI and ISA signaling. Miscellaneous system interface terminals include multifunction terminals: $\overline{\text{SUSPEND}}$, $\overline{\text{RI_OUT}}$/PME (power-management control signal), and SPKROUT.



NOTE: The PC Card interface is 68 terminals for CardBus and 16-bit PC Cards. In ZV mode, 23 terminals are used for routing the ZV signals to the VGA controller and audio subsystem.

Figure 3–1. PCI1410A Simplified Block Diagram

3.1 Power-Supply Sequencing

The PCI1410A device contains 3.3-V I/O buffers with 5-V tolerance, requiring a core power supply and clamp voltages. The core power supply always is 3.3 V. The clamp voltages can be either 3.3 V or 5 V, depending on the interface. The following power-up and power-down sequences are recommended.

The power-up sequence is:

1. Apply 3.3-V power to the core.
2. Assert $\overline{\text{GRST}}$ to the device to disable the outputs during power up. Output drivers must be powered up in the high-impedance state to prevent high current levels through the clamp diodes to the 5-V supply.
3. Apply the clamp voltage.

The power-down sequence is:

1. Use $\overline{\text{GRST}}$ to switch outputs to a high-impedance state.
2. Remove the clamp voltage.
3. Remove the 3.3-V power from the core.

3.2 I/O Characteristics

Figure 3–2 shows a 3-state bidirectional buffer. Section 8.2, *Recommended Operating Conditions*, provides the electrical characteristics of the inputs and outputs.

NOTE: The PCI1410A device meets the ac specifications of the *PC Card Standard* and *PCI Local Bus Specification.*

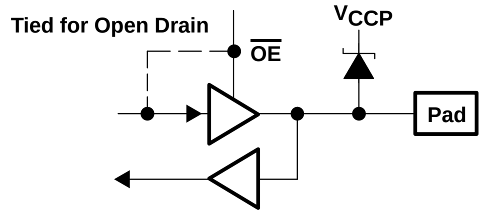


Figure 3–2. 3-State Bidirectional Buffer

NOTE: Unused pins (input or I/O) must be held high or low to prevent them from floating.

3.3 Clamping Voltages

The clamping voltages are set to match whatever external environment the PCI1410A device is interfaced with: 3.3 V or 5 V. The I/O sites can be pulled through a clamping diode to a voltage rail that protects the core from external signals. The core power supply always is 3.3 V and is independent of the clamping voltages. For example, PCI signaling can be either 3.3 V or 5 V, and the PCI1410A device must reliably accommodate both voltage levels. This is accomplished by using a 3.3-V I/O buffer that is 5-V tolerant, with the applicable clamping voltage applied. If a system designer desires a 5-V PCI bus, V_{CCP} can be connected to a 5-V power supply.

The PCI1410A device requires three separate clamping voltages because it supports a wide range of features. The three voltages are listed and defined in Section 8.2, *Recommended Operating Conditions*.

3.4 Peripheral Component Interconnect (PCI) Interface

The PCI1410A device is fully compliant with the *PCI Local Bus Specification*. The PCI1410A device provides all required signals for PCI master or slave operation, and can operate in either a 5-V or 3.3-V signaling environment by connecting the V_{CCP} terminals to the desired voltage level. In addition to the mandatory PCI signals, the PCI1410A device provides the optional interrupt signal $\overline{INTA}$.

3.4.1 PCI Bus Lock ($\overline{LOCK}$)

The bus-locking protocol defined in the *PCI Local Bus Specification* is not highly recommended, but is provided on the PCI1410A device as an additional compatibility feature. The PCI $\overline{LOCK}$ signal can be routed to the MFUNC4 terminal via the multifunction routing register. See Section 4.30, *Multifunction Routing Register*, for details. Note that the use of $\overline{LOCK}$ is supported only by PCI-to-CardBus bridges in the downstream direction (away from the processor).

PCI $\overline{LOCK}$ indicates an atomic operation that may require multiple transactions to complete. When $\overline{LOCK}$ is asserted, nonexclusive transactions can proceed to an address that currently is not locked. A grant to start a transaction on the PCI bus does not guarantee control of $\overline{LOCK}$; control of $\overline{LOCK}$ is obtained under its own protocol. It is possible for different initiators to use the PCI bus while a single master retains ownership of $\overline{LOCK}$. Note that the CardBus signal for this protocol is $\overline{CBLOCK}$ to avoid confusion with the bus clock.

An agent may need to do an exclusive operation because a critical access to memory might be broken into several transactions, but the master wants exclusive rights to a region of memory. The granularity of the lock is defined by PCI to be 16 bytes, aligned. The $\overline{LOCK}$ protocol defined by the *PCI Local Bus Specification* allows a resource lock without interfering with nonexclusive real-time data transfer, such as video.

The PCI bus arbiter may be designed to support only complete bus locks using the $\overline{LOCK}$ protocol. In this scenario, the arbiter will not grant the bus to any other agent (other than the $\overline{LOCK}$ master) while $\overline{LOCK}$ is asserted. A complete

bus lock may have a significant impact on the performance of the video. The arbiter that supports complete bus lock must grant the bus to the cache to perform a writeback due to a snoop to a modified line when a locked operation is in progress.

The PCI1410A device supports all $\overline{\text{LOCK}}$ protocol associated with PCI-to-PCI bridges, as also defined for PCI-to-CardBus bridges. This includes disabling write posting while a locked operation is in progress, which can solve a potential deadlock when using devices such as PCI-to-PCI bridges. The potential deadlock can occur if a CardBus target supports delayed transactions and blocks access to the target until it completes a delayed read. This target characteristic is prohibited by the *PCI Local Bus Specification*, and the issue is resolved by the PCI master using $\overline{\text{LOCK}}$.

3.4.2 Loading Subsystem Identification

The subsystem vendor ID register (PCI offset 40h, see Section 4.26) and subsystem ID register (PCI offset 42h, see Section 4.27) make up a doubleword of PCI configuration space located at offset 40h for functions 0 and 1. This doubleword register is used for system and option card (mobile dock) identification purposes and is required by some operating systems.

The PCI1410A device offers two mechanisms to load a read-only value into the subsystem registers. The first mechanism relies upon the system BIOS providing the subsystem ID value. The default access mode to the subsystem registers is read-only, but can be made read/write by clearing bit 5 (SUBSYSRW) in the system control register (PCI offset 80h, see Section 4.29). When this bit is cleared, the BIOS can write a subsystem identification value into the registers at PCI offset 40h. The BIOS must set the SUBSYSRW bit such that the subsystem vendor ID register and subsystem ID register are limited to read-only access. This approach saves the added cost of implementing the serial electrically erasable programmable ROM (EEPROM).

In some conditions, such as in a docking environment, the subsystem vendor ID register and subsystem ID register must be loaded with a unique identifier via a serial EEPROM. The PCI1410A device loads the data from the serial EEPROM after a reset of the primary bus. Note that the $\overline{\text{SUSPEND}}$ input gates the PCI reset from the entire PCI1410A core, including the serial bus state machine (see Section 3.8.4, *Suspend Mode*, for details on using $\overline{\text{SUSPEND}}$).

The PCI1410A device provides a two-line serial bus host controller that can interface to a serial EEPROM. See Section 3.6, *Serial Bus Interface*, for details on the two-wire serial bus controller and applications.

3.5 PC Card Applications

This section describes the PC Card interfaces of the PCI1410A device:

- Card insertion/removal and recognition
- P^{2}C power-switch interface
- Zoomed-video (ZV) support
- Speaker and audio applications
- LED socket activity indicators
- PC Card-16 DMA support
- PC Card controller programming model
- CardBus socket registers

3.5.1 PC Card Insertion/Removal and Recognition

The *PC Card Standard* addresses the card-detection and recognition process through an interrogation procedure that the socket must initiate on card insertion into a cold, nonpowered socket. Through this interrogation, card voltage requirements and interface (16-bit versus CardBus) are determined.

The scheme uses the card-detect and voltage-sense signals. The configuration of these four terminals identifies the card type and voltage requirements of the PC Card interface. The encoding scheme is defined in the *PC Card Standard* and in Table 3–1.

Table 3–1. PC Card Card-Detect and Voltage-Sense Connections

CD2//CCD2	CD1//CCD1	VS2//CVS2	VS1//CVS1	KEY	INTERFACE	VOLTAGE
Ground	Ground	Open	Open	5 V	16-bit PC Card	5 V
Ground	Ground	Open	Ground	5 V	16-bit PC Card	5 V and 3.3 V
Ground	Ground	Ground	Ground	5 V	16-bit PC Card	5 V, 3.3 V, and X.X V
Ground	Ground	Open	Ground	LV	16-bit PC Card	3.3 V
Ground	Connect to CVS1	Open	Connect to CCD1	LV	CardBus PC Card	3.3 V
Ground	Ground	Ground	Ground	LV	16-bit PC Card	3.3 V and X.X V
Connect to CVS2	Ground	Connect to CCD2	Ground	LV	CardBus PC Card	3.3 V and X.X V
Connect to CVS1	Ground	Ground	Connect to CCD2	LV	CardBus PC Card	3.3 V, X.X V, and Y.Y V
Ground	Ground	Ground	Open	LV	16-bit PC Card	Y.Y V
Connect to CVS2	Ground	Connect to CCD2	Open	LV	CardBus PC Card	Y.Y V
Ground	Connect to CVS2	Connect to CCD1	Open	LV	CardBus PC Card	X.X V and Y.Y V
Connect to CVS1	Ground	Open	Connect to CCD2	LV	CardBus PC Card	Y.Y V
Ground	Connect to CVS1	Ground	Connect to CCD1	Reserved		
Ground	Connect to CVS2	Connect to CCD1	Ground	Reserved		

3.5.2 P2C Power-Switch Interface (TPS2211)

The PCI1410A device provides a P2C (PCMCIA peripheral control) interface for control of the PC Card power switch. The VCCD and VPPD terminals are used with the TI TPS2211 single-slot PC Card power-switch interface to provide power-switch support. Figure 3–3 shows terminal assignments for the TPS2211 power-switch interface. Figure 3–4 illustrates a typical application, where the PCI1410A device represents the PC Card controller.

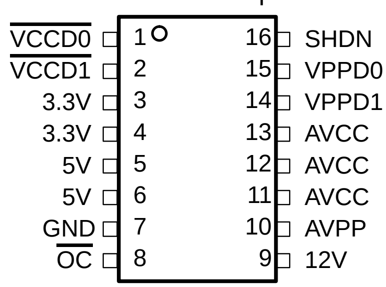


Figure 3–3. TPS2211 Terminal Assignments

The PCI1410A device also includes support for the Maxim™ 1602 single-channel CardBus and PCMCIA power-switching network. Application of this power switch is similar to the TPS2211 power-switch interface.

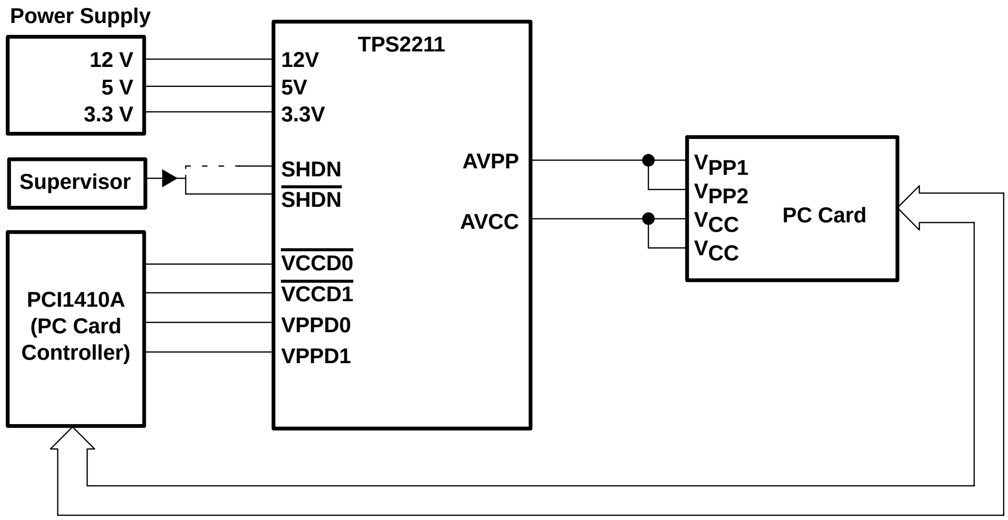


Figure 3–4. TPS2211 Typical Application

3.5.3 Zoomed-Video Support

The PCI1410A device allows for the implementation of ZV for PC Cards. ZV is supported by setting bit 6 (ZVENABLE) in the card control register (PCI offset 91h, see Section 4.32) on a per-socket function basis. Setting this bit puts PC Card 16 address lines ADDR25–ADDR4 of the PC Card interface in the high-impedance state. These lines can then transfer video and audio data directly to the appropriate controller. Card address lines ADDR3–ADDR0 can still access PC Card CIS registers for PC Card configuration. Figure 3–5 illustrates a PCI1410A ZV implementation.

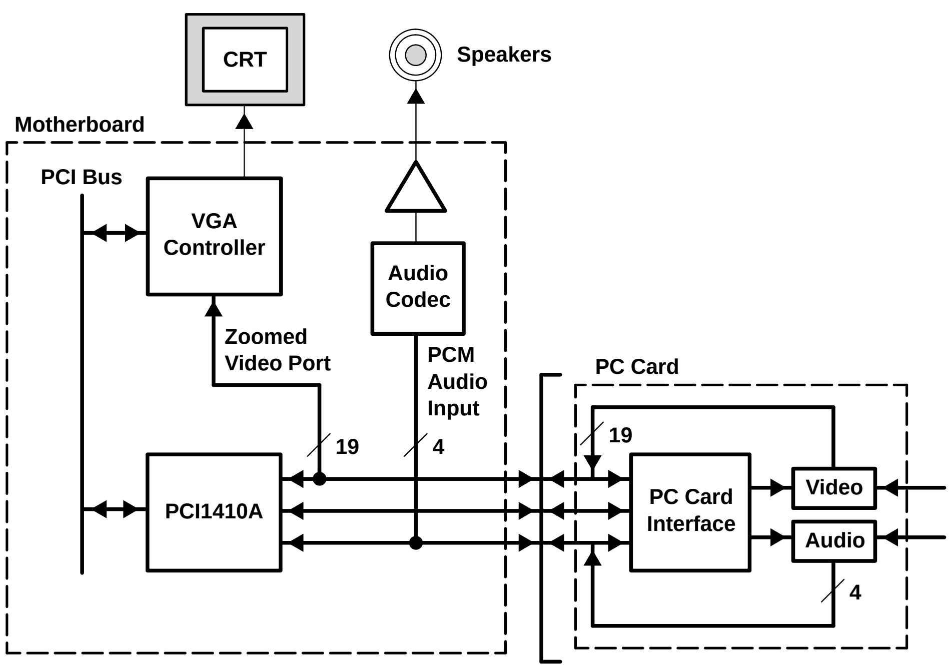


Figure 3–5. Zoomed-Video Implementation Using the PCI1410A Device

Not shown in Figure 3–5 is the multiplexing scheme used to route a socket ZV source or an external ZV source to the graphics controller. A typical external source might be provided from a high-speed serial bus like IEEE 1394. The PCI1410A device provides ZVSTAT and ZVSEL0 signals on the multifunction terminals to switch external bus drivers. Figure 3–6 shows an implementation for switching between two ZV streams using external logic.

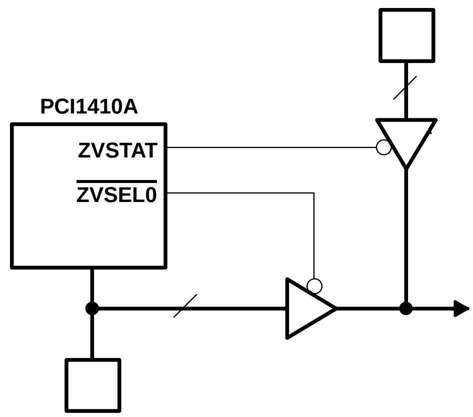


Figure 3–6. Zoomed-Video Switching Application

Figure 3–6 illustrates an implementation using standard three-state bus drivers with active-low output enables. ZVSEL0 is an active-low output indicating that the socket ZV mode is enabled. ZVSTAT is an active-high output, indicating that the PCI1410A socket is enabled for ZV mode. The implementation shown in Figure 3–6 can be used if PC Card ZV is prioritized over other sources.

3.5.4 Ultrazoomed Video

Ultrazoomed video is an enhancement to the PCI1410A DMA engine and is intended to improve the 16-bit bandwidth for MPEG I and MPEG II decoder PC Cards. This enhancement allows the PCI1410A device to fetch 32 bits of data from memory, versus the PCI11XX/PCI12XX 16-bit fetch capability. This enhancement allows a higher sustained throughput to the 16-bit PC Card because the PCI1410A device prefetches an extra 16 bits (32 bits total) during each PCI read transaction. If the PCI bus becomes busy, the PCI1410A device has an extra 16 bits of data to perform back-to-back 16-bit transactions to the PC Card before having to fetch more data. This feature is built into the DMA engine, and software is not required to enable this enhancement.

> **NOTE:** The PCI11XX and PCI12XX series CardBus controllers have enough 16-bit bandwidth to support MPEG II PC Card decoders. But, it was decided to improve the bandwidth even more in the PCI14XX series CardBus controllers.

3.5.5 Internal Ring Oscillator

The internal ring oscillator provides an internal clock option for the PCI1410A device so that the PCI clock is not required for the PCI1410A device to power down a socket or interrogate a PC Card. This internal oscillator operates nominally at 16 kHz and can be enabled by setting bit 27 (P2CCLK) of the system control register (PCI offset 80h, see Section 4.29) to 1. This function is disabled by default.

3.5.6 Integrated Pullup Resistors for PC Card Interface

The *PC Card Standard* requires pullup resistors on various terminals to support both CardBus and 16-bit card configurations. Unlike the PCI1210 or PCI1211 device, which required external pullup resistors, the PCI1410A device has integrated all of these pullup resistors.

SIGNAL NAME	PIN NUMBER		
	GGU	PGE	GHK
ADDR14/CPERR	C13	104	F14
READY/CINT	D6	132	A10
ADDR15/CIRDY	A12	110	C15
CD1/CCD1	L12	75	L19
VS1/CVS1	C6	131	F11
ADDR19/CBLOCK	D11	103	E19
ADDR20/CSTOP	C12	105	E18
ADDR21/CDEVSEL	B13	107	E17
ADDR22/CTRDY	A13	109	A16
VS2/CVS2	D9	117	E13
RESET/CRST	B9	119	F12
WAIT/CSERR	A5	133	B10
INPACK/CREQ	B8	123	E12
BVD2(SPKR)/CAUDIO	B5	134	C10
BVD1(STSCHG)/CSTSCHG	C5	135	E10
CD2/CCD2	A4	137	A9

3.5.7 SPKROUT and CAUDPWM Usage

SPKROUT carries the digital audio signal from the PC Card to the system. When a 16-bit PC Card is configured for I/O mode, the BVD2 terminal becomes SPKR. This terminal also is used in CardBus binary audio applications, and is referred to as CAUDIO. SPKR passes a TTL-level digital audio signal to the PCI1410A device. The CardBus CAUDIO signal also can pass a single-amplitude binary waveform. The binary audio signal from the PC Card socket is used in the PCI1410A device to produce SPKROUT. This output is enabled by bit 1 (SPKROUTEN) in the card control register (PCI offset 91h, see Section 4.32).

Older controllers support CAUDIO in binary or PWM mode, but use the same terminal (SPKROUT). Some audio chips may not support both modes on one terminal and may have a separate terminal for binary and PWM. The PCI1410A implementation includes a signal for PWM, CAUDPWM, which can be routed to an MFUNC terminal. Bit 2 (AUD2MUX), located in the card control register, is programmed to route a CardBus CAUDIO PWM terminal to CAUDPWM. See Section 4.30, *Multifunction Routing Register*, for details on configuring the MFUNC terminals.

Figure 3–7 illustrates a sample application using SPKROUT and CAUDPWM.

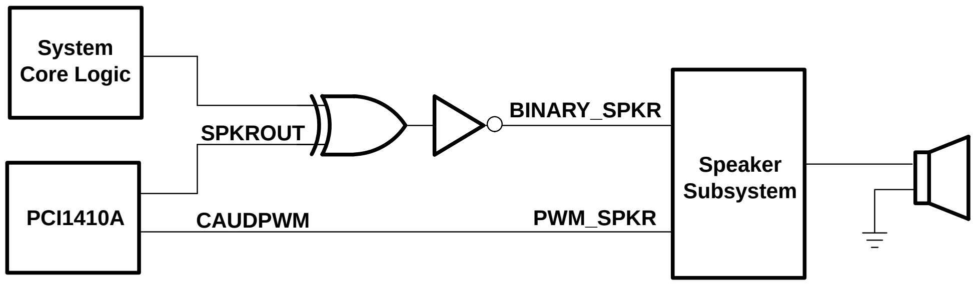


Figure 3–7. Sample Application of SPKROUT and CAUDPWM

3.5.8 LED Socket Activity Indicators

The socket activity LEDs indicate when a PC Card is being accessed. The LED_SKT signal can be routed to the multifunction terminals. When configured for LED output, this terminal outputs an active high signal to indicate socket activity. See Section 4.30, *Multifunction Routing Register*, for details on configuring the multifunction terminals.

The LED signal is active high and is driven in pulses of 64-ms duration. When the LED is not being driven high, it is driven to a low state. Either of the two circuits shown in Figure 3–8 can be implemented to provide LED signaling, and it is left for the board designer to implement the circuit that best fits the application.

The LED activity signals are valid when a card is inserted, powered, and not in reset. For PC Card-16, the LED activity signal is pulsed when READY/$\overline{\text{IREQ}}$ is low. For CardBus cards, the LED activity signal is pulsed if $\overline{\text{CFRAME}}$, $\overline{\text{CIRDY}}$, or $\overline{\text{CREQ}}$ is active.

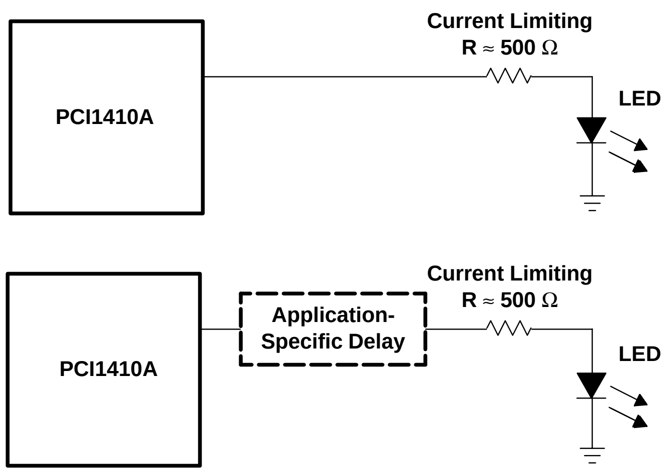


Figure 3–8. Two Sample LED Circuits

As indicated, the LED signals are driven for a period of 64 ms by a counter circuit. To avoid the possibility of the LED appearing to be stuck when the PCI clock is stopped, the LED signaling is cut off when the $\overline{\text{SUSPEND}}$ signal is asserted, when the PCI clock is to be stopped during the clock run protocol, or when in the D2 or D1 power state.

If any additional socket activity occurs during this counter cycle, the counter is reset and the LED signal remains driven. If socket activity is frequent (at least once every 64 ms), the LED signal remains driven.

3.5.9 PC Card-16 Distributed DMA Support

The PCI1410A device supports a distributed DMA slave engine for 16-bit PC Card DMA support. The distributed DMA (DDMA) slave register set provides the programmability necessary for the slave DMA engine. Table 3–2 provides the DMA register configuration.

Two socket function-dependent PCI configuration header registers that are critical for DMA are the socket DMA register 0 (PCI offset 94h, see Section 4.35) and the socket DMA register 1 (PCI offset 98h, see Section 4.36).

Distributed DMA is enabled through socket DMA register 0, and the contents of this register configure the PC Card-16 terminal (SPKR, IOIS16, or INPACK), which is used for the DMA request signal, DREQ. The base address of the DMA slave registers and the transfer size (bytes or words) are programmed through the socket DMA register 1. See the programming model and register descriptions in Section 7 for details.

Table 3–2. Distributed DMA Registers

<table>
<tr><th>TYPE</th><th colspan="4">REGISTER NAME</th><th>DMA
BASE ADDRESS
OFFSET</th></tr>
<tr><td>R</td><td rowspan="2">Reserved</td><td rowspan="2">Page</td><td colspan="2">Current address</td><td rowspan="2">00h</td></tr>
<tr><td>W</td><td colspan="2">Base address</td></tr>
<tr><td>R</td><td rowspan="2">Reserved</td><td rowspan="2">Reserved</td><td colspan="2">Current count</td><td rowspan="2">04h</td></tr>
<tr><td>W</td><td colspan="2">Base count</td></tr>
<tr><td>R</td><td>N/A</td><td rowspan="2">Reserved</td><td>N/A</td><td>Status</td><td rowspan="2">08h</td></tr>
<tr><td>W</td><td>Mode</td><td>Request</td><td>Command</td></tr>
<tr><td>R</td><td>Multichannel</td><td rowspan="2">Reserved</td><td>N/A</td><td rowspan="2">Reserved</td><td rowspan="2">0Ch</td></tr>
<tr><td>W</td><td>Mask</td><td>Master clear</td></tr>
</table>

The DMA registers contain control and status information consistent with the 8237 DMA controller; however, the register locations are reordered and expanded in some cases. While the DMA register definitions are identical to those of the same name in the 8237 DMA controller, some register bits defined in the 8237 DMA controller do not apply to distributed DMA in a PCI environment. In such cases, the PCI1410A device implements these obsolete register bits as read-only, nonfunctional bits. The reserved registers shown in Table 3–2 are implemented as read-only and return 0s when read. Write transactions to reserved registers have no effect.

The DMA transfer is prefaced by several configuration steps that are specific to the PC Card and must be completed after the PC Card is inserted and interrogated. These steps include setting the proper DREQ signal assignment, setting the data transfer width, and mapping and enabling the DMA register set. As discussed above, this is done through socket DMA register 0 and socket DMA register 1. The DMA register set is then programmed similarly to an 8237 controller, and the PCI1410A device awaits a DREQ assertion from the PC Card requesting a DMA transfer.

DMA writes transfer data from the PC Card-to-PCI memory addresses. The PCI1410A device accepts data 8 or 16 bits at a time, depending on the programmed data width, and then requests access to the PCI bus by asserting its REQ signal. Once the PCI bus is granted in an idle state, the PCI1410A device initiates a PCI memory write command to the current memory address and transfers the data in a single data phase. After terminating the PCI cycle, the PCI1410A device accepts the next byte(s) from the PC Card until the transfer count expires.

DMA reads transfer data from PCI memory addresses to the PC Card application. Upon the assertion of DREQ, the PCI1410A device asserts REQ to acquire the PCI bus. Once the bus is granted in an idle state, the PCI1410A device initiates a PCI memory read operation to the current memory address and accepts 8 or 16 bits of data, depending on the programmed data width. After terminating the PCI cycle, the data is passed on to the PC Card. After terminating the PC Card cycle, the PCI1410A device requests access to the PCI bus again, until the transfer count has expired.

The PCI1410A target interface acts normally during this procedure and accepts I/O reads and writes to the DMA registers. While a DMA transfer is in progress and the host resets the DMA channel, the PCI1410A device asserts TC and ends the PC Card cycle(s). TC is indicated in the DMA status register (DMA offset 08h, see Section 7.5). At the PC Card interface, the PCI1410A device supports demand mode transfers. The PCI1410A device asserts DACK during the transfer unless DREQ is deasserted before TC. TC is mapped to the OE PC Card terminal for DMA write operations and is mapped to the WE PC Card terminal for DMA read operations. The DACK signal is mapped to the PC Card REG signal in all transfers, and the DREQ terminal is routed to one of three options, which is programmed through socket DMA register 0.

3.5.10 PC Card-16 PC/PCI DMA

Some chip sets provide a way for legacy I/O devices to do DMA transfers on the PCI bus. In the PC/PCI DMA protocol, the PCI1410A device acts as a PCI target device to certain DMA-related I/O addresses. The PCI1410A $\overline{\text{PCREQ}}$ and $\overline{\text{PCGNT}}$ signals are provided as a point-to-point connection to a chipset supporting PC/PCI DMA. The $\overline{\text{PCREQ}}$ and $\overline{\text{PCGNT}}$ signals may be routed to the MFUNC2 and MFUNC5 terminals, respectively. See Section 4.30, *Multifunction Routing Register*, for details on configuring the multifunction terminals.

Under the PC/PCI protocol, a PCI DMA slave device (such as the PCI1410A device) requests a DMA transfer on a particular channel using a serialized protocol on $\overline{\text{PCREQ}}$. The I/O DMA bus master arbitrates for the PCI bus and grants the channel through a serialized protocol on $\overline{\text{PCGNT}}$ when it is ready for the transfer. The I/O cycle and memory cycles are then presented on the PCI bus, which performs the DMA transfers similarly to legacy DMA master devices.

PC/PCI DMA is enabled for each PC Card-16 slot by setting bit 19 (CDREQEN) in the respective system control register (PCI offset 80h, see Section 4.29). On power up, this bit is reset and the card PC/PCI DMA is disabled. Bit 3 (CDMA_EN) of the system control register is a global enable for PC/PCI DMA, and is set at power up and never cleared if the PC/PCI DMA mechanism is implemented. The desired DMA channel for each PC Card-16 slot must be configured through bits 18–16 (CDMACHAN field) in the system control register. The channels are configured as indicated in Table 3–3.

Table 3–3. PC/PCI Channel Assignments

SYSTEM CONTROL REGISTER			DMA CHANNEL	CHANNEL TRANSFER DATA WIDTH
BIT 18	BIT 17	BIT16		
0	0	0	Channel 0	8-bit DMA transfers
0	0	1	Channel 1	8-bit DMA transfers
0	1	0	Channel 2	8-bit DMA transfers
0	1	1	Channel 3	8-bit DMA transfers
1	0	0	Channel 4	Not used
1	0	1	Channel 5	16-bit DMA transfers
1	1	0	Channel 6	16-bit DMA transfers
1	1	1	Channel 7	16-bit DMA transfers

As in distributed DMA, the PC Card terminal mapped to $\overline{\text{DREQ}}$ must be configured through socket DMA register 0 (PCI offset 94h, see Section 4.35). The data transfer width is a function of channel number, and the DMA slave registers are not used. When a $\overline{\text{DREQ}}$ is received from a PC Card and the channel has been granted, the PCI1410A device decodes the I/O addresses listed in Table 3–4 and performs actions dependent upon the address.

Table 3–4. I/O Addresses Used for PC/PCI DMA

DMA I/O ADDRESS	DMA CYCLE TYPE	TERMINAL COUNT	PCI CYCLE TYPE
00h	Normal	0	I/O read/write
04h	Normal TC	1	I/O read/write
C0h	Verify	0	I/O read
C4h	Verify TC	1	I/O read

When the PC/PCI DMA is used as a PC Card-16 DMA mechanism, it may not provide the performance levels of DMA; however, the design of a PCI target implementing PC/PCI DMA is considerably less complex. No bus master state machine is required to support PC/PCI DMA, since the DMA control is centralized in the chipset. This DMA scheme often is referred to as centralized DMA for this reason.

3.5.11 CardBus Socket Registers

The PCI1410A device contains all registers for compatibility with the *PC Card Standard*. These registers exist as the CardBus socket registers and are listed in Table 3–5.

Table 3–5. CardBus Socket Registers

REGISTER NAME	OFFSET
Socket event	00h
Socket mask	04h
Socket present state	08h
Socket force event	0Ch
Socket control	10h
Reserved	14h
Reserved	18h
Reserved	1Ch
Socket power management	20h

3.6 Serial Bus Interface

The PCI1410A device provides a serial bus interface to load subsystem identification and select register defaults through a serial EEPROM and to provide a PC Card power switch interface alternative to P^{2}C. See Section 3.5.2, *P^{2}C Power-Switch Interface (TPS2211)*, for details. The PCI1410A serial bus interface is compatible with various I^{2}C and SMBus components.

3.6.1 Serial Bus-Interface Implementation

To enable the serial interface, a pullup resistor must be implemented on the $\overline{\text{VCCD0}}$ and $\overline{\text{VCCD1}}$ terminals and the appropriate pullup resistors must be implemented on the SDA and SCL signals, that is, the MFUNC1 and MFUNC4 terminals. When the interface is detected, bit 3 (SBDETECT) in the serial bus control and status register (PCI offset B3h, see Section 4.50) is set. The SBDETECT bit is cleared by a writeback of 1.

The PCI1410A device implements a two-pin serial interface with one clock signal (SCL) and one data signal (SDA). When pullup resistors are provided on the $\overline{\text{VCCD0}}$ and $\overline{\text{VCCD1}}$ terminals, the SCL signal is mapped to the MFUNC4 terminal and the SDA signal is mapped to the MFUNC1 terminal. The PCI1410A device drives SCL at nearly 100 kHz during data transfers, which is the maximum specified frequency for standard-mode I^{2}C. The serial EEPROM must be located at address A0h. Figure 3–9 illustrates an example application implementing the two-wire serial bus.

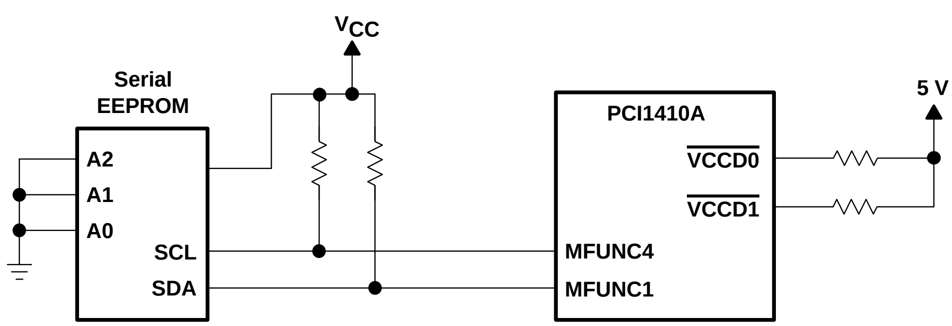


Figure 3–9. Serial EEPROM Application

Some serial device applications may include PC Card power switches, ZV source switches, card ejectors, or other devices that may enhance the user’s PC Card experience. The serial EEPROM device and PC Card power switches are discussed in the sections that follow.

3.6.2 Serial Bus-Interface Protocol

The SCL and SDA signals are bidirectional, open-drain signals and require pullup resistors as shown in Figure 3–9. The PCI1410A device supports up to 100-Kb/s data transfer rate and is compatible with standard-mode I^{2}C using 7-bit addressing.

All data transfers are initiated by the serial bus master. The beginning of a data transfer is indicated by a start condition, which is signaled when the SDA line transitions to a low state while SCL is in the high state, as illustrated in Figure 3–10. The end of a requested data transfer is indicated by a stop condition, which is signaled by a low-to-high transition of SDA while SCL is in the high state, as shown in Figure 3–10. Data on SDA must remain stable during the high state of the SCL signal, as changes on the SDA signal during the high state of SCL are interpreted as control signals, that is, a start or a stop condition.

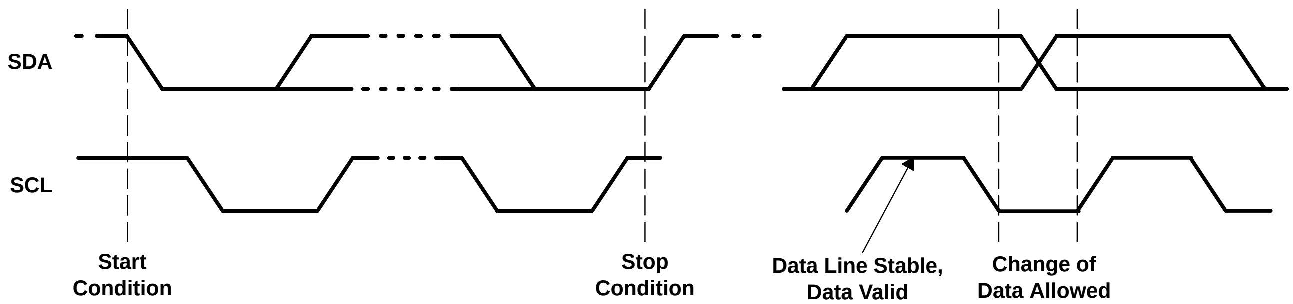


Figure 3–10. Serial Bus Start/Stop Conditions and Bit Transfers

Data is transferred serially in 8-bit bytes. The number of bytes that can be transmitted during a data transfer is unlimited; however, each byte must be completed with an acknowledge bit. An acknowledge (ACK) is indicated by the receiver pulling the SDA signal low so that it remains low during the high state of the SCL signal. Figure 3–11 illustrates the acknowledge protocol.

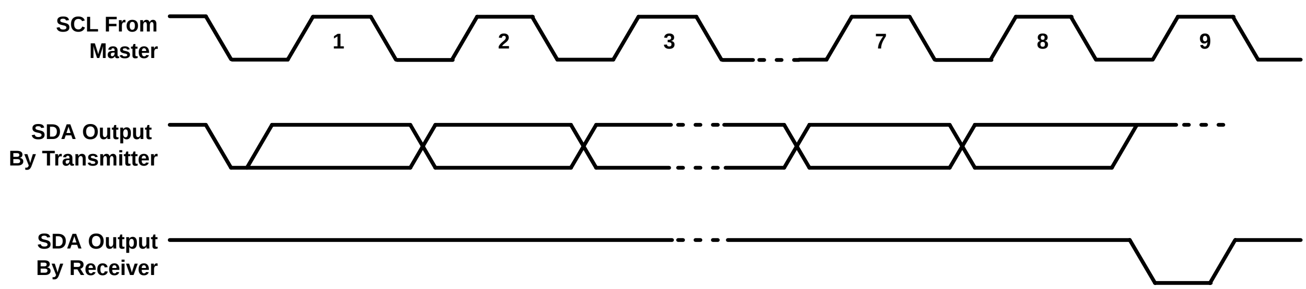


Figure 3–11. Serial Bus-Protocol Acknowledge

The PCI1410A device is a serial bus master; all other devices connected to the serial bus external to the PCI1410A device are slave devices. As the bus master, the PCI1410A device drives the SCL clock at nearly 100 kHz during bus cycles and places SCL in a high-impedance state (zero frequency) during idle states.

Typically, the PCI1410A device masters byte reads and byte writes under software control. Doubleword reads are performed by the serial EEPROM initialization circuitry upon a PCI reset, and may not be generated under software control. See Section 3.6.3, *Serial Bus EEPROM Application*, for details on how the PCI1410A device automatically loads the subsystem identification and other register defaults through a serial bus EEPROM.

Figure 3–12 illustrates a byte write. The PCI1410A device issues a start condition and sends the 7-bit slave device address and the command bit 0. A 0 in the R/$\overline{W}$ command bit indicates that the data transfer is a write. The slave device acknowledges if it recognizes the address. If there is no acknowledgment received by the PCI1410A device, an appropriate status bit is set in the serial bus control and status register (PCI offset B3h, see Section 4.50). The word address byte is then sent by the PCI1410A device and another slave acknowledgment is expected. The PCI1410A device then delivers the data byte, MSB first, and expects a final acknowledgment before issuing the stop condition.

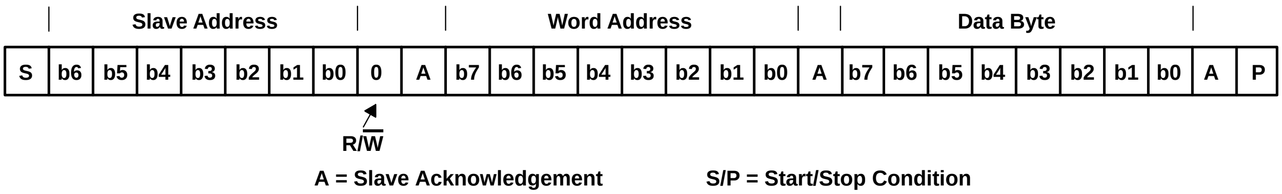


Figure 3–12. Serial Bus Protocol – Byte Write

Figure 3–13 illustrates a byte read. The read protocol is very similar to the write protocol, except the R/$\overline{W}$ command bit must be set to 1 to indicate a read-data transfer. In addition, the PCI1410A master must acknowledge reception of the read bytes from the slave transmitter. The slave transmitter drives the SDA signal during read data transfers. The SCL signal remains driven by the PCI1410A master.

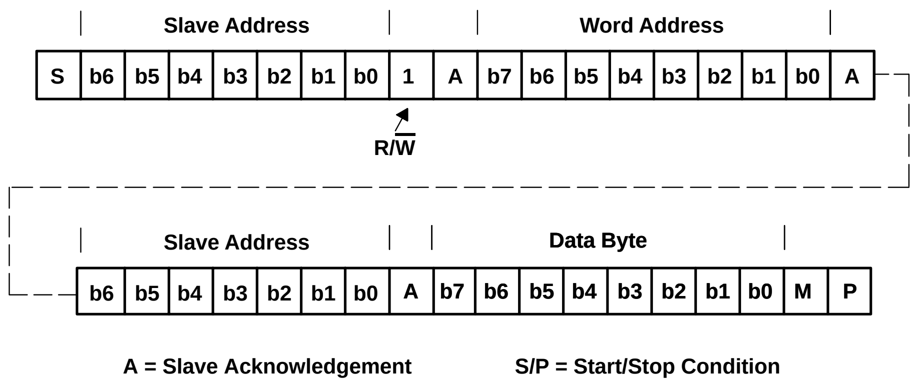


Figure 3–13. Serial Bus Protocol – Byte Read

Figure 3–14 illustrates EEPROM interface doubleword data-collection protocol.

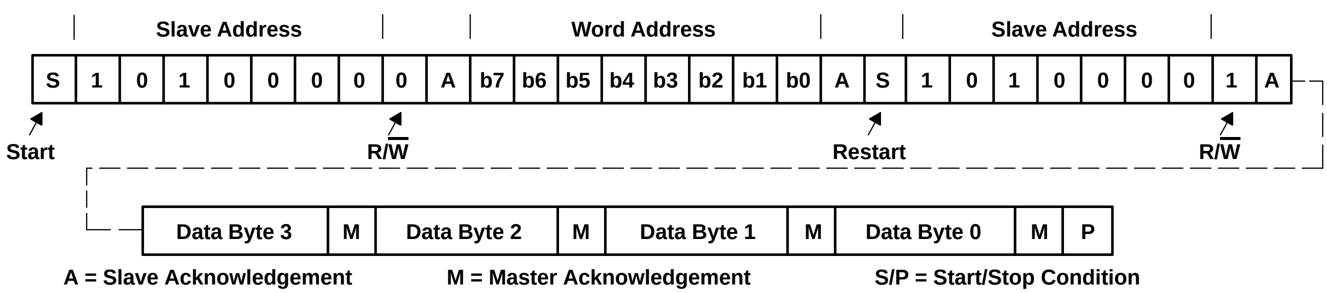


Figure 3–14. EEPROM Interface Doubleword Data Collection

3.6.3 Serial Bus EEPROM Application

When the PCI bus is reset and the serial bus interface is detected, the PCI1410A device attempts to read the subsystem identification and other register defaults from a serial EEPROM. The registers and corresponding bits that may be loaded with defaults through the EEPROM are provided in Table 3–6.

Table 3–6. Registers and Bits Loadable Through Serial EEPROM

OFFSET REFERENCE	PCI OFFSET	REGISTER	BITS LOADED FROM EEPROM
01h	40h	Subsystem vendor ID, subsystem ID	31–0
02h	80h	System control	31–30, 27, 26, 24, 15, 14, 6–3, 1
03h	8Ch	Multifunction routing	27–0
04h	90h	Retry status, card control, device control, diagnostic	31, 28–24, 22, 19–16, 15, 7, 6

Figure 3–15 details the EEPROM data format. This format must be followed for the PCI1410A device to properly load initializations from a serial EEPROM. Any undefined condition results in a terminated load and sets bit 0 (ROM_ERR) in the serial bus control and status register (PCI offset B3h, see Section 4.50).

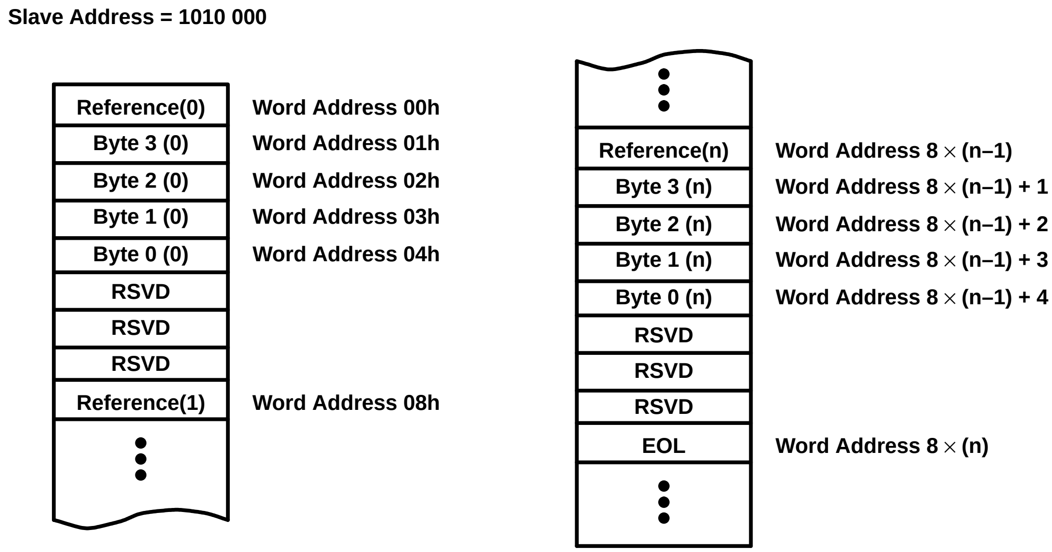


Figure 3–15. EEPROM Data Format

The byte at the EEPROM word address 00h must contain either a valid offset reference, as listed in Table 3–6, or an end-of-list (EOL) indicator. The EOL indicator has a byte value of FFh, and indicates the end of the data to load from the EEPROM. Only doubleword registers are loaded from the EEPROM, and all bit fields must be considered when programming the EEPROM.

The serial EEPROM is addressed at slave address 1010 000b by the PCI1410A device. All hardware address bits for the EEPROM should be tied to the appropriate level to achieve this address. The serial EEPROM chip in the sample application circuit (Figure 3–9) assumes the 1010b high address nibble. The lower three address bits are terminal inputs to the chip, and the sample application shows these terminal inputs tied to GND.

When a valid offset reference is read, four bytes are read from the EEPROM, MSB first, as illustrated in Figure 3–14. The address autoincrements after every byte transfer according to the doubleword read protocol. Note that the word addresses align with the data format illustrated in Figure 3–15. The PCI1410A device continues to load data from the serial EEPROM until an end-of-list indicator is read. Three reserved bytes are stuffed to maintain 8-byte data structures.

Note, the 8-byte data structure is important to provide correct addressing per the doubleword read format shown in Figure 3–14. In addition, the reference offsets must be loaded in the EEPROM in sequential order, that is, 01h, 02h, 03h, 04h. If the offsets are not sequential, the registers will be loaded incorrectly.

3.6.4 Accessing Serial Bus Devices Through Software

The PCI1410A device provides a programming mechanism to control serial bus devices through software. The programming is accomplished through a doubleword of PCI configuration space at offset B0h. Table 3–7 lists the registers used to program a serial bus device through software.

Table 3–7. PCI1410A Registers Used to Program Serial Bus Devices

PCI OFFSET	REGISTER NAME	DESCRIPTION
B0h	Serial bus data	Contains the data byte to send on write commands or the received data byte on read commands
B1h	Serial bus index	The content of this register is sent as the word address on byte writes or reads. This register is not used in the quick command protocol.
B2h	Serial bus slave address	Write transactions to this register initiate a serial bus transaction. The slave device address and the R/$\overline{W}$ command selector are programmed through this register.
B3h	Serial bus control and status	Read data valid, general busy, and general error status are communicated through this register. In addition, the protocol select bit is programmed through this register.

To write a byte, the serial bus data register must be programmed with the data, the serial bus index register must be programmed with the byte address, and the serial bus slave address register must be programmed with the 7-bit slave address (SLAVADDR field) and bit 0 (RWCMD) must be reset.

On byte reads, the byte address is programmed into the serial bus index register, the serial bus slave address register must be programmed with the 7-bit slave address (SLAVADDR field) and bit 0 (RWCMD) must be set, and bit 5 (REQBUSY) in the serial bus control and status register (PCI offset B3h, see Section 4.50) must be polled until clear. Then the contents of the serial bus data register are valid read data from the serial bus interface.

3.7 Programmable Interrupt Subsystem

Interrupts provide a way for I/O devices to let the microprocessor know that they require servicing. The dynamic nature of PC Cards and the abundance of PC Card I/O applications require substantial interrupt support from the PCI1410A device. The PCI1410A device provides several interrupt signaling schemes to accommodate the needs of a variety of platforms. The different mechanisms for dealing with interrupts in this device are based on various specifications and industry standards. The ExCA register set provides interrupt control for some 16-bit PC Card functions, and the CardBus socket register set provides interrupt control for the CardBus PC Card functions. The PCI1410A device is, therefore, backward compatible with existing interrupt control register definitions, and new registers have been defined where required.

The PCI1410A device detects PC Card interrupts and events at the PC Card interface and notifies the host controller, using one of several interrupt signaling protocols. To simplify the discussion of interrupts in the PCI1410A device, PC Card interrupts are classified either as card status change (CSC) or as functional interrupts.

The method by which any type of PCI1410A interrupt is communicated to the host interrupt controller varies from system to system. The PCI1410A device offers system designers the choice of using parallel PCI interrupt signaling, parallel ISA-type IRQ interrupt signaling, or the IRQSER serialized ISA and/or PCI interrupt protocol. It is possible to use the parallel PCI interrupts in combination with either parallel IRQs or serialized IRQs, as detailed in the sections that follow. All interrupt signaling is provided through the seven multifunction terminals, MFUNC0–MFUNC6.

3.7.1 PC Card Functional and Card Status Change Interrupts

PC Card functional interrupts are defined as requests from a PC Card application for interrupt service and are indicated by asserting specially defined signals on the PC Card interface. Functional interrupts are generated by 16-bit I/O PC Cards and by CardBus PC Cards.

Card status change (CSC)-type interrupts are defined as events at the PC Card interface that are detected by the PCI1410A device, and may warrant notification of host card and socket services software for service. CSC events include both card insertion and removal from PC Card sockets, as well as transitions of certain PC Card signals.

Table 3–8 summarizes the sources of PC Card interrupts and the type of card associated with them. CSC and functional interrupt sources are dependent on the type of card inserted in the PC Card socket. The three types of cards that can be inserted into any PC Card socket are:

- 16-bit memory cards
- 16-bit I/O cards
- CardBus cards

Functional interrupt events are valid only for 16-bit I/O and CardBus cards; that is, the functional interrupts are not valid for 16-bit memory cards. Furthermore, card insertion and removal-type CSC interrupts are independent of the card type.

Table 3–8. PC Card Interrupt Events and Description

CARD TYPE	EVENT	TYPE	SIGNAL	DESCRIPTION
16-bit memory	Battery conditions (BVD1, BVD2)	CSC	BVD1($\overline{\text{STSCHG}}$)//CSTSCHG	A transition on BVD1 indicates a change in the PC Card battery conditions.
			BVD2($\overline{\text{SPKR}}$)//CAUDIO	A transition on BVD2 indicates a change in the PC Card battery conditions.
	Wait states (READY)	CSC	READY($\overline{\text{IREQ}}$)//$\overline{\text{CINT}}$	A transition on READY indicates a change in the ability of the memory PC Card to accept or provide data.
16-bit I/O	Change in card status (STSCHG)	CSC	BVD1($\overline{\text{STSCHG}}$)//CSTSCHG	The assertion of $\overline{\text{STSCHG}}$ indicates a status change on the PC Card.
	Interrupt request (IREQ)	Functional	READY($\overline{\text{IREQ}}$)//$\overline{\text{CINT}}$	The assertion of $\overline{\text{IREQ}}$ indicates an interrupt request from the PC Card.
CardBus	Change in card status (CSTSCHG)	CSC	BVD1($\overline{\text{STSCHG}}$)//CSTSCHG	The assertion of CSTSCHG indicates a status change on the PC Card.
	Interrupt request ($\overline{\text{CINT}}$)	Functional	READY($\overline{\text{IREQ}}$)//$\overline{\text{CINT}}$	The assertion of $\overline{\text{CINT}}$ indicates an interrupt request from the PC Card.
All PC Cards	Card insertion or removal	CSC	$\overline{\text{CD1}}$//$\overline{\text{CCD1}}$, $\overline{\text{CD2}}$//$\overline{\text{CCD2}}$	A transition on either $\overline{\text{CD1}}$//$\overline{\text{CCD1}}$ or $\overline{\text{CD2}}$//$\overline{\text{CCD2}}$ indicates an insertion or removal of a 16-bit or CardBus PC Card.
	Power cycle complete	CSC	N/A	An interrupt is generated when a PC Card power-up cycle has completed.

The naming convention for PC Card signals describes the function for 16-bit memory, I/O cards, and CardBus. For example, READY($\overline{\text{IREQ}}$)//$\overline{\text{CINT}}$ includes READY for 16-bit memory cards, $\overline{\text{IREQ}}$ for 16-bit I/O cards, and $\overline{\text{CINT}}$ for CardBus cards. The 16-bit memory card signal name is first, with the I/O card signal name second, enclosed in parentheses. The CardBus signal name follows after a double slash (//).

The *PC Card Standard* describes the power-up sequence that must be followed by the PCI1410A device when an insertion event occurs and the host requests that the socket V_{CC} and V_{PP} be powered. Upon completion of this power-up sequence, the PCI1410A interrupt scheme can be used to notify the host system (see Table 3–8), denoted by the power cycle complete event. This interrupt source is considered a PCI1410A internal event because it depends on the completion of applying power to the socket rather than on a signal change at the PC Card interface.

3.7.2 Interrupt Masks and Flags

Host software may individually mask (or disable) most of the potential interrupt sources listed in Table 3–8 by setting the appropriate bits in the PCI1410A device. By individually masking the interrupt sources listed, software can control those events that cause a PCI1410A interrupt. Host software has some control over the system interrupt the PCI1410A device asserts by programming the appropriate routing registers. The PCI1410A device allows host software to route PC Card CSC and PC Card functional interrupts to separate system interrupts. Interrupt routing is somewhat specific to the interrupt signaling method used. This is discussed in more detail in the following sections.

When an interrupt is signaled by the PCI1410A device, the interrupt service routine must determine which of the events listed in Table 3–9 caused the interrupt. Internal registers in the PCI1410A device provide flags that report the source of an interrupt. By reading these status bits, the interrupt service routine can determine the action to be taken.

Table 3–9 details the registers and bits associated with masking and reporting potential interrupts. All interrupts can be masked, except the functional PC Card interrupts, and an interrupt status flag is available for all types of interrupts.

Table 3–9. Interrupt Mask and Flag Registers

CARD TYPE	EVENT	MASK	FLAG
16-bit memory	Battery conditions (BVD1, BVD2)	ExCA offset 05h/805h bits 1 and 0	ExCA offset 04h/804h bits 1 and 0
	Wait states (READY)	ExCA offset 05h/805h bit 2	ExCA offset 04h/804h bit 2
16-bit I/O	Change in card status (STSCHG)	ExCA offset 05h/805h bit 0	ExCA offset 04h/804h bit 0
	Interrupt request (IREQ)	Always enabled	PCI configuration offset 91h bit 0
All 16-bit PC Cards	Power cycle complete	ExCA offset 05h/805h bit 3	ExCA offset 04h/804h bit 3
CardBus	Change in card status (CSTSCHG)	Socket mask bit 0	Socket event bit 0
	Interrupt request (CINT)	Always enabled	PCI configuration offset 91h bit 0
	Power cycle complete	Socket mask bit 3	Socket event bit 3
	Card insertion or removal	Socket mask bits 2 and 1	Socket event bits 2 and 1

Notice that there is not a mask bit to stop the PCI1410A device from passing PC Card functional interrupts through to the appropriate interrupt scheme. These interrupts are not valid until the card is properly powered, and there never should be a card interrupt that does not require service after proper initialization.

Table 3–9 lists the various methods of clearing the interrupt flag bits. The flag bits in the ExCA registers (16-bit PC Card-related interrupt flags) can be cleared using two different methods. One method is an explicit write of 1 to the flag bit to clear, and the other is by reading the flag bit register. The selection of flag bit clearing is made by bit 2 (IFCMODE) in the ExCA global control register (ExCA offset 1Eh, see Section 5.20) and defaults to the *flag cleared on read* method.

The CardBus-related interrupt flags can be cleared by an explicit write of 1 to the interrupt flag in the socket force event register (CardBus offset 0Ch, see Section 6.4). Although some of the functionality is shared between the CardBus registers and the ExCA registers, software should not program the chip through both register sets when a CardBus card is functioning.

3.7.3 Using Parallel IRQ Interrupts

The seven multifunction terminals, MFUNC6–MFUNC0, implemented in the PCI1410A device may be routed to obtain a subset of the ISA IRQs . The IRQ choices provide ultimate flexibility in PC Card host interruptions. To use the parallel ISA type IRQ interrupt signaling, software must program the device control register (PCI offset 92h, see Section 4.33), to select the parallel IRQ signaling scheme. See Section 4.30, *Multifunction Routing Register*, for details on configuring the multifunction terminals.

A system using parallel IRQs requires a minimum of one PCI terminal, INTA, to signal CSC events. This requirement is dictated by certain card and socket services software. The INTA requirement calls for routing the MFUNC0 terminal for INTA signaling. This leaves (at a maximum) six different IRQs to support legacy 16-bit PC Card functions.

As an example, suppose the six IRQs used by legacy PC Card applications are IRQ3, IRQ4, IRQ5, IRQ10, IRQ11, and IRQ15. The multifunction routing register must be programmed to a value of 0FBA 5432h. This value routes the MFUNC0 terminal to INTA signaling and routes the remaining terminals as illustrated in Figure 3–16. Not shown is that INTA also must be routed to the programmable interrupt controller (PIC), or to some circuitry that provides parallel PCI interrupts to the host.

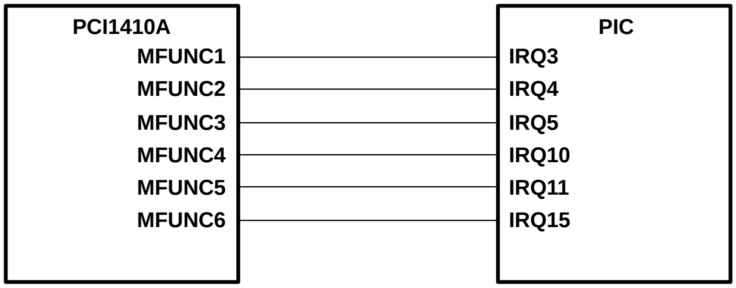


Figure 3–16. IRQ Implementation

Power-on software is responsible for programming the multifunction routing register to reflect the IRQ configuration of a system implementing the PCI1410A device. See Section 4.30, *Multifunction Routing Register*, for details on configuring the multifunction terminals.

The parallel ISA-type IRQ signaling from the MFUNC6–MFUNC0 terminals is compatible with those input directly into the 8259 PIC. The parallel IRQ option is provided for system designs that require legacy ISA IRQs. Design constraints may demand more MFUNC6–MFUNC0 IRQ terminals than the PCI1410A device makes available.

3.7.4 Using Parallel PCI Interrupts

Parallel PCI interrupts are available when exclusively in parallel PCI interrupt mode, parallel ISA IRQ signaling mode, and when only IRQs are serialized with the IRQSER protocol. The socket function interrupts are routed to INTA (MFUNC0).

3.7.5 Using Serialized IRQSER Interrupts

The serialized interrupt protocol implemented in the PCI1410A device uses a single terminal to communicate all interrupt status information to the host controller. The protocol defines a serial packet consisting of a start cycle, multiple interrupt indication cycles, and a stop cycle. All data in the packet is synchronous with the PCI clock. The packet data describes 16 parallel ISA IRQ signals and the optional 4 PCI interrupts INTA, INTB, INTC, and INTD. For details on the IRQSER protocol, refer to the document *Serialized IRQ Support for PCI Systems*.

3.7.6 SMI Support in the PCI1410A Device

The PCI1410A device provides a mechanism for interrupting the system when power changes have been made to the PC Card socket interfaces. The interrupt mechanism is designed to fit into a system maintenance interrupt (SMI) scheme. SMI interrupts are generated by the PCI1410A device, when enabled, after a write cycle to either the socket control register (CardBus offset 10h, see Section 6.5) of the CardBus register set or the ExCA power control register (ExCA offset 02h, see Section 5.3) causes a power cycle change sequence to be sent on the power-switch interface.

The SMI control is programmed through three bits in the system control register (PCI offset 80h, see Section 4.29). These bits are SMIROUTE (bit 26), SMISTATUS (bit 25), and SMIENB (bit 24). Table 3–10 describes the SMI control bits function.

Table 3–10. SMI Control

BIT NAME	FUNCTION
SMIROUTE	This shared bit controls whether the SMI interrupts are sent as a CSC interrupt or as IRQ2.
SMISTATUS	This socket-dependent bit is set when an SMI interrupt is pending. This status flag is cleared by writing back a 1.
SMIENB	When set, SMI interrupt generation is enabled.

If CSC SMI interrupts are selected, the SMI interrupt is sent as the CSC. The CSC interrupt can be either level or edge mode, depending upon bit 1 (CSCMODE) in the ExCA global control register (ExCA offset 1Eh, see Section 5.20).

If IRQ2 is selected by SMIROUTE, the IRQSER signaling protocol supports SMI signaling in the IRQ2 IRQ/Data slot. In a parallel ISA IRQ system, the support for an active low IRQ2 is provided only if IRQ2 is routed to either MFUNC1, MFUNC3, or MFUNC6 through the multifunction routing register (PCI offset 8Ch, see Section 4.30).

3.8 Power-Management Overview

In addition to the low-power CMOS technology process used for the PCI1410A device, various features are designed into the device to allow implementation of popular power-saving techniques. These features and techniques are discussed in this section.

3.8.1 Clock-Run Protocol

The PCI CLKRUN feature is the primary method of power management on the PCI interface of the PCI1410A device. CLKRUN signaling is provided through the MFUNC6 terminal. Because some chipsets do not implement CLKRUN, this is not always available to the system designer, and alternate power-saving features are provided. For details on the CLKRUN protocol see the *PCI Mobile Design Guide*.

The PCI1410A device does not permit the central resource to stop the PCI clock under any of the following conditions:

- Bit 1 (KEEPCLK) in the system control register (PCI offset 80h, see Section 4.29) is set.
- The PC Card-16 resource manager is busy.
- The PCI1410A CardBus master state machine is busy. A cycle may be in progress on CardBus.
- The PCI1410A master is busy. There may be posted data from CardBus to PCI in the PCI1410A device.
- Interrupts are pending.
- The CardBus CCLK for either socket has not been stopped by the PCI1410A CCLKRUN manager.

The PCI1410A device restarts the PCI clock using the CLKRUN protocol under any of the following conditions:

- A PC Card-16 IREQ or a CardBus CINT has been asserted.
- A CardBus CBWAKE (CSTSCHG) or PC Card-16 STSCHG/RI event occurs.
- A CardBus attempts to start the CCLK using CCLKRUN.
- A CardBus card arbitrates for the CardBus bus using CREQ.
- A 16-bit DMA PC Card asserts DREQ.

3.8.2 CardBus PC Card Power Management

The PCI1410A device implements its own card power-management engine that can turn off CCLK to a socket when there is no activity to the CardBus PC Card. The PCI clock-run protocol is followed on the CardBus CCLKRUN interface to control this clock management.

3.8.3 16-Bit PC Card Power Management

Bit 7 (COE) in the ExCA power control register (ExCA offset 02h, see Section 5.3) and bit 0 (PWRDWN) in the ExCA global control register (ExCA offset 1Eh, see Section 5.20) are provided for 16-bit PC Card power management. The COE bit places the card interface in a high-impedance state to save power. The power savings when using this feature are minimal. The COE bit will reset the PC Card when used, and the PWRDWN bit will not. Furthermore, the PWRDWN bit is an automatic COE, that is, the PWRDWN performs the COE function when there is no card activity.

NOTE: The 16-bit PC Card must implement the proper pullup resistors for the COE and PWRDWN modes.

3.8.4 Suspend Mode

The SUSPEND signal, provided for backward compatibility, gates the PRST (PCI reset) signal and the GRST (global reset) signal from the PCI1410A device. Besides gating PRST and GRST, SUSPEND also gates PCLK inside the PCI1410A device to minimize power consumption.

Gating PCLK does not create any issues with respect to the power-switch interface in the PCI1410A device. This is because the PCI1410A device does not depend on the PCI clock to clock the power-switch interface. There are two methods to clock the power-switch interface in the PCI1410A device:

- Use an external clock to the PCI1410A PCLK terminal.
- Use the internal oscillator.

It also should be noted that asynchronous signals, such as card status-change interrupts and RI_OUT, can be passed to the host system without a PCI clock. However, if card status-change interrupts are routed over the serial interrupt stream, the PCI clock must be restarted to pass the interrupt, because neither the internal oscillator nor an external clock is routed to the serial interrupt state machine. Figure 3–17 shows the suspend logic diagram.

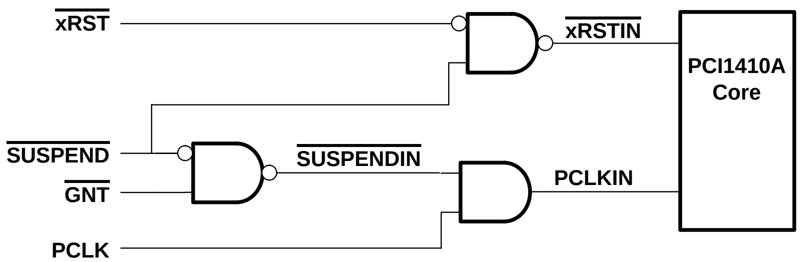


Figure 3–17. Suspend Logic Diagram

Figure 3–18 is a signal diagram of the suspend function.

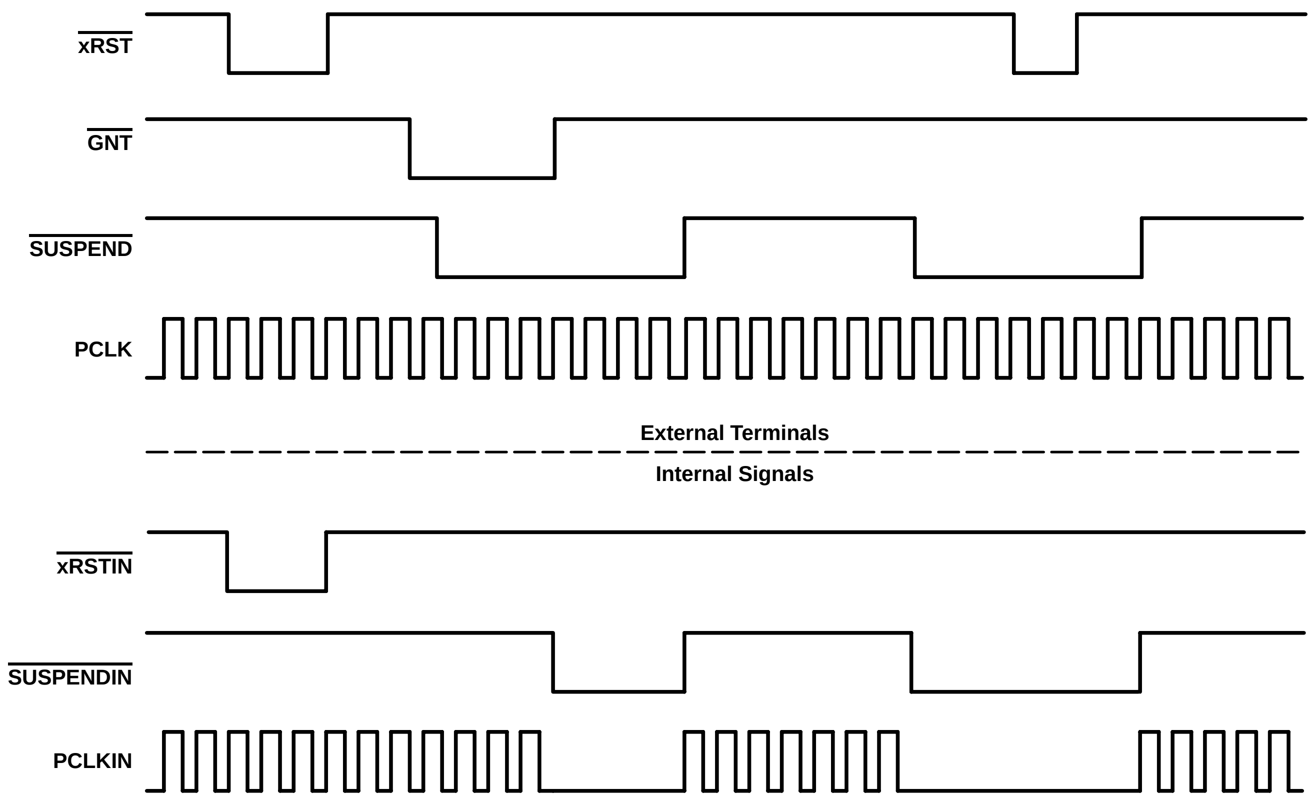


Figure 3–18. Signal Diagram of Suspend Function

3.8.5 Requirements for Suspend Mode

The suspend mode prevents the clearing of all register contents on the assertion of reset (PRST or GRST) that would require the reconfiguration of the PCI1410A device by software. Asserting the SUSPEND signal places the controller PCI outputs in a high-impedance state and gates the PCLK signal internally to the controller unless a PCI transaction currently is in process (GNT is asserted). It is important that the PCI bus not be parked on the PCI1410A device when SUSPEND is asserted, because the outputs are in a high-impedance state.

The GPIOs, MFUNC signals, and RI_OUT signals are all active during SUSPEND, unless they are disabled in the appropriate PCI1410A registers.

3.8.6 Ring Indicate

The RI_OUT output is an important feature in power management, allowing a system to go into a suspended mode and wake up on modem rings and other card events. TI-designed flexibility permits this signal to fit wide platform requirements. RI_OUT on the PCI1410A device can be asserted under any of the following conditions:

- A 16-bit PC Card modem in a powered socket asserts RI to indicate to the system the presence of an incoming call.
- A powered-down CardBus card asserts CSTSCHG (CBWAKE), requesting system and interface wake-up.
- A powered CardBus card asserts CSTSCHG from the insertion/removal of cards or change in battery voltage levels.

Figure 3–19 shows various enable bits for the PCI1410A RI_OUT function; however, it does not show the masking of CSC events. See Table 3–9 for a detailed description of CSC interrupt masks and flags.

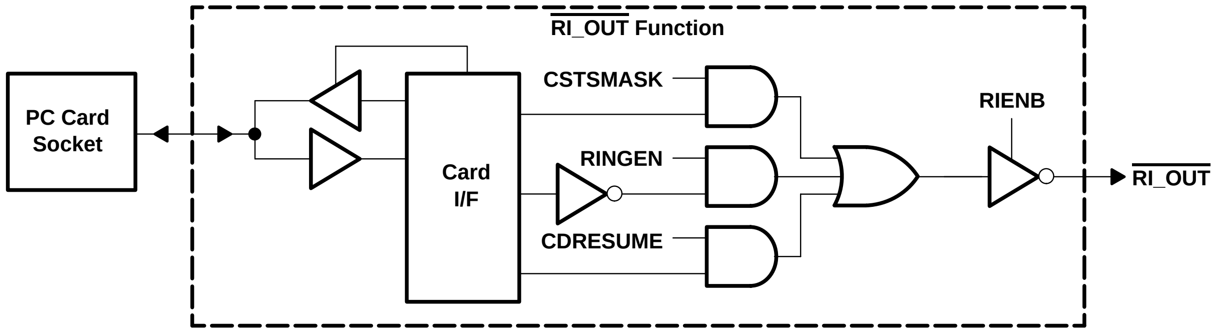


Figure 3–19. RI_OUT Functional Diagram

RI from the 16-bit PC Card interface is masked by bit 7 (RINGEN) in the ExCA interrupt and general control register (ExCA offset 03h, see Section 5.4). This is programmed on a per-socket basis and is applicable only when a 16-bit card is powered in the socket.

The CBWAKE, signaling to RI_OUT, is enabled through the same mask as the CSC event for CSTSCHG. The mask bit (bit 0, CSTSMASK) is programmed through the socket mask register (CardBus offset 04h, see Section 6.2) in the CardBus socket registers.

3.8.7 PCI Power Management

The *PCI Bus Power Management Interface Specification for PCI to CardBus Bridges* establishes the infrastructure required for the operating system to control the power of PCI functions. This is done by defining a standard PCI interface and operations to manage the power of PCI functions on the bus. The PCI bus and the PCI functions can be assigned one of four software-visible power-management states that result in varying levels of power savings.

The four power-management states of PCI functions are:

- D0 – Fully-on state
- D1 and D2 – Intermediate states
- D3 – Off state

Similarly, bus power states of the PCI bus are B0–B3. The bus power states B0–B3 are derived from the device power state of the originating bridge device.

For the operating system (OS) to manage the device power states on the PCI bus, the PCI function must support four power-management operations. These operations are:

- Capabilities reporting
- Power status reporting
- Setting the power state
- System wake-up

The OS identifies the capabilities of the PCI function by traversing the new capabilities list. The presence of capabilities, in addition to the standard PCI capabilities, is indicated by a 1 in bit 4 (CAPLIST) of the status register (PCI offset 06h, see Section 4.5).

The capabilities pointer provides access to the first item in the linked list of capabilities. For the PCI1410A device, a CardBus bridge with PCI configuration space header type 2, the capabilities pointer is mapped to an offset of 14h. The first byte of each capability register block is required to be a unique ID of that capability. PCI power management has been assigned an ID of 01h. The next byte is a pointer to the next pointer item in the list of capabilities. If there are no more items in the list, the next item pointer should be set to 0. The registers following the next item pointer are specific to the capabilities of their corresponding power-management functions. The PCI power-management capability implements the register block outlined in Table 3–11.

Table 3–11. Power-Management Registers

<table>
<tr><th colspan="4">REGISTER NAME</th><th>OFFSET</th></tr>
<tr><td colspan="2">Power management capabilities</td><td>Next-item pointer</td><td>Capability ID</td><td>A0h</td></tr>
<tr><td>PM data</td><td>PMCSR bridge support extensions</td><td colspan="2">Power management control/status</td><td>A4h</td></tr>
</table>

The power management capabilities register (PCI offset A2h, see Section 4.39) provides information on the capabilities of the function related to power management. The power management control/status register (PCI offset A4h, see Section 4.40) enables control of power management states and enables/monitors power management events. The data register is an optional register that can provide dynamic data.

For more information on PCI power management, see the *PCI Bus Power Management Interface Specification for PCI to CardBus Bridges*.

3.8.8 CardBus Device Class Power Management

The *PCI Bus Power Management Interface Specification for PCI to CardBus Bridges* was approved by PCMCIA in December of 1997. This specification follows the device and bus state definitions provided in the *PCI Bus Power Management Interface Specification* published by the PCI Special Interest Group (SIG). The main issue addressed in the *PCI Bus Power Management Interface Specification for PCI to CardBus Bridges* is wake-up from $D3_{hot}$ or $D3_{cold}$ without losing wake-up context (also called $\overline{\text{PME}}$ context).

The specific issues addressed by the *PCI Bus Power Management Interface Specification for PCI to CardBus Bridges* for D3 wake up are as follows:

- Preservation of device context: The specification states that a reset must occur when transitioning from D3 to D0. Some method to preserve wake-up context must be implemented so that the reset does not clear the $\overline{\text{PME}}$ context registers.
- Power source in $D3_{cold}$ if wake-up support is required from this state.

The Texas Instruments PCI1410A device addresses these D3 wake-up issues in the following manner:

- Two resets are provided to handle preservation of $\overline{\text{PME}}$ context bits:
  - Global reset ($\overline{\text{GRST}}$) is used only on the initial boot up of the system after power up. It places the PCI1410A device in its default state and requires BIOS to configure the device before becoming fully functional.
  - PCI reset ($\overline{\text{PRST}}$) now has dual functionality based on whether $\overline{\text{PME}}$ is enabled or not. If $\overline{\text{PME}}$ is enabled, $\overline{\text{PME}}$ context is preserved. If $\overline{\text{PME}}$ is not enabled, $\overline{\text{PRST}}$ acts the same as a normal PCI reset. Please see the master list of $\overline{\text{PME}}$ context bits in Section 3.8.10.
- Power source in $D3_{cold}$ if wake-up support is required from this state. Since V_{CC} is removed in $D3_{cold}$, an auxiliary power source must be supplied to the PCI1410A V_{CC} terminals. Consult the *PCI14xx Implementation Guide for D3 Wake-Up* or the *PCI Power Management Interface Specification for PCI to CardBus Bridges* for further information.

3.8.9 ACPI Support

The *Advanced Configuration and Power Interface (ACPI) Specification* provides a mechanism that allows unique pieces of hardware to be described to the ACPI driver. The PCI1410A device offers a generic interface that is compliant with ACPI design rules.

Two doublewords of general-purpose ACPI programming bits reside in PCI1410A PCI configuration space at offset A8h. The programming model is broken into status and control functions. In compliance with ACPI, the top-level event status and enable bits reside in the general-purpose event status (PCI offset A8h, see Section 4.43) and general-purpose event enable (PCI offset AAh, see Section 4.44) registers. The status and enable bits are implemented as defined by ACPI and illustrated in Figure 3–20.

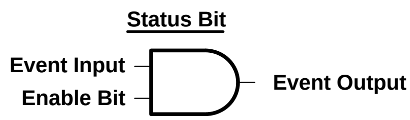


Figure 3–20. Block Diagram of a Status/Enable Cell

The status and enable bits generate an event that allows the ACPI driver to call a control method associated with the pending status bit. The control method can then control the hardware by manipulating the hardware control bits or by investigating child status bits and calling their respective control methods. A hierarchical implementation would be somewhat limiting, however, as upstream devices would have to remain in some level of power state to report events.

For more information on ACPI, see the *Advanced Configuration and Power Interface (ACPI) Specification.*

3.8.10 Master List of PME Context Bits and Global Reset-Only Bits

If the PME enable bit (bit 8) of the power management control/status register (PCI offset A4h, see Section 4.40) is asserted, the assertion of PRST will not clear the following PME context bits. If the PME enable bit is not asserted, the PME context bits are cleared with PRST. The PME context bits are:

- Bridge control register (PCI offset 3Eh, see Section 4.25): bit 6
- Power management capabilities register (PCI offset A2h, see Section 4.39): bit 15
- Power management control/status register (PCI offset A4h, see Section 4.40): bits 15, 8
- ExCA power control register (ExCA offset 802h, see Section 5.3): bits 4, 3, 1, 0
- ExCA interrupt and general control (ExCA offset 803h, see Section 5.4): bits 6, 5
- ExCA card status-change-interrupt configuration register (ExCA offset 805h, see Section 5.6): bits 3–0
- CardBus socket event register (CardBus offset 00h, see Section 6.1): bits 3–0
- CardBus socket mask register (CardBus offset 04h, see Section 6.2): bits 3–0
- CardBus socket present state register (CardBus offset 08h, see Section 6.3): bits 13–10, 7, 5–0
- CardBus socket control register (CardBus offset 10h, see Section 6.5): bits 6–4, 2–0

Global reset places all registers in their default state, regardless of the state of the PME enable bit. The GRST signal is gated only by the SUSPEND signal. This means that assertion of SUSPEND blocks the GRST signal internally, thus preserving all register contents. The registers cleared by GRST are:

- Subsystem vendor ID (PCI offset 40h, see Section 4.26): bits 15–0
- Subsystem ID (PCI offset 42h, see Section 4.27): bits 15–0
- PC Card 16-bit legacy-mode base address register (PCI offset 44h, see Section 4.28): bits 31–1
- System control register (PCI offset 80h, see Section 4.29): bits 31, 30, 27, 26, 24–14, 7–0
- Multifunction routing register (PCI offset 8Ch, see Section 4.30): bits 27–0
- Retry status register (PCI offset 90h, see Section 4.31): bits 7, 6, 3, 1
- Card control register (PCI offset 91h, see Section 4.32): bits 7–5, 2–0
- Device control register (PCI offset 92h, see Section 4.33): bits 7–5, 3–0
- Diagnostic register (PCI offset 93h, see Section 4.34): bits 7–0
- Socket DMA register 0 (PCI offset 94h, see Section 4.35): bits 1–0
- Socket DMA register 1 (PCI offset 98h, see Section 4.36): bits 15–4, 2–0
- General-purpose event enable register (PCI offset AAh, see Section 4.44): bits 15, 11, 8, 4–0
- General-purpose output (PCI offset AEh, see Section 4.46): bits 4–0
- Serial bus data (PCI offset B0h, see Section 4.47): bits 7–0
- Serial bus index (PCI offset B1h, see Section 4.48): bits 7–0
- Serial bus slave address register (PCI offset B2h, see Section 4.49): bits 7–0
- Serial bus control and status register (PCI offset B3h, see Section 4.50): bits 7, 2
- ExCA identification and revision register (ExCA offset 00h, see Section 5.1): bits 7–0
- ExCA card status change register (ExCA offset 804h, see Section 5.5): bits 3–0
- ExCA global control register (ExCA offset 1Eh, see Section 5.20): bits 3–0

4 PC Card Controller Programming Model

This section describes the PCI1410A PCI configuration registers that make up the 256-byte PCI configuration header for each PCI1410A function. As noted, some bits are global in nature and are accessed only through function 0.

4.1 PCI Configuration Registers

The configuration header is compliant with the *PCI Local Bus Specification* as a CardBus bridge header and is *PC 99* compliant as well. Table 4–1 shows the PCI configuration header, which includes both the predefined portion of the configuration space and the user-definable registers.

Table 4–1. PCI Configuration Registers

<table>
<tr><th colspan="4">REGISTER NAME</th><th>OFFSET</th></tr>
<tr><td colspan="2">Device ID</td><td colspan="2">Vendor ID</td><td>00h</td></tr>
<tr><td colspan="2">Status</td><td colspan="2">Command</td><td>04h</td></tr>
<tr><td colspan="3">PCI class code</td><td>Revision ID</td><td>08h</td></tr>
<tr><td>BIST</td><td>Header type</td><td>Latency timer</td><td>Cache line size</td><td>0Ch</td></tr>
<tr><td colspan="4">CardBus socket/ExCA base-address</td><td>10h</td></tr>
<tr><td colspan="2">Secondary status</td><td>Reserved</td><td>Capability pointer</td><td>14h</td></tr>
<tr><td>CardBus latency timer</td><td>Subordinate bus number</td><td>CardBus bus number</td><td>PCI bus number</td><td>18h</td></tr>
<tr><td colspan="4">Memory base register 0</td><td>1Ch</td></tr>
<tr><td colspan="4">Memory limit register 0</td><td>20h</td></tr>
<tr><td colspan="4">Memory base register 1</td><td>24h</td></tr>
<tr><td colspan="4">Memory limit register 1</td><td>28h</td></tr>
<tr><td colspan="4">I/O base register 0</td><td>2Ch</td></tr>
<tr><td colspan="4">I/O limit register 0</td><td>30h</td></tr>
<tr><td colspan="4">I/O base register 1</td><td>34h</td></tr>
<tr><td colspan="4">I/O limit register 1</td><td>38h</td></tr>
<tr><td colspan="2">Bridge control</td><td>Interrupt pin</td><td>Interrupt line</td><td>3Ch</td></tr>
<tr><td colspan="2">Subsystem ID</td><td colspan="2">Subsystem vendor ID</td><td>40h</td></tr>
<tr><td colspan="4">PC Card 16-bit I/F legacy-mode base address</td><td>44h</td></tr>
<tr><td colspan="4">Reserved</td><td>48h–7Ch</td></tr>
<tr><td colspan="4">System control</td><td>80h</td></tr>
<tr><td colspan="4">Reserved</td><td>84h–88h</td></tr>
<tr><td colspan="4">Multifunction routing</td><td>8Ch</td></tr>
<tr><td>Diagnostic</td><td>Device control</td><td>Card control</td><td>Retry status</td><td>90h</td></tr>
<tr><td colspan="4">Socket DMA register 0</td><td>94h</td></tr>
<tr><td colspan="4">Socket DMA register 1</td><td>98h</td></tr>
<tr><td colspan="4">Reserved</td><td>9Ch</td></tr>
<tr><td colspan="2">Power management capabilities</td><td>Next-item pointer</td><td>Capability ID</td><td>A0h</td></tr>
<tr><td>Power management data</td><td>Power management control/status bridge support extensions</td><td colspan="2">Power management control/status</td><td>A4h</td></tr>
<tr><td colspan="2">General-purpose event enable</td><td colspan="2">General-purpose event status</td><td>A8h</td></tr>
<tr><td colspan="2">General-purpose output</td><td colspan="2">General-purpose input</td><td>ACh</td></tr>
<tr><td>Serial bus control and status</td><td>Serial bus slave address</td><td>Serial bus index</td><td>Serial bus data</td><td>B0h</td></tr>
<tr><td colspan="4">Reserved</td><td>B4h–FCh</td></tr>
</table>

A bit description table, typically included when a register contains bits of more than one type or purpose, indicates bit field names, a detailed field description, and field access tags, which appear in the type column of the bit-description table. Table 4–2 describes the field access tags.

Table 4–2. Bit-Field Access Tag Descriptions

ACCESS TAG	NAME	MEANING
R	Read	Field can be read by software.†
W	Write	Field can be written by software to any value.
S	Set	Field can be set by a write of 1. Writes of 0 have no effect.
C	Clear	Field can be cleared by a write of 1. Writes of 0 have no effect.
U	Update	Field can be autonomously updated by the PCI1410A device.

† A bit can display either of two types of behavior when read. After having been read, it can maintain the value it had previously, or the read process can cause it to be reset to 0.

4.2 Vendor ID Register

The vendor ID register contains a value allocated by the PCI Special Interest Group (SIG) and identifies the manufacturer of the PCI device. The vendor ID assigned to TI is 104Ch.

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Vendor ID															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	1	0	0	0	0	0	1	0	0	1	1	0	0

Register: **Vendor ID**
Offset: 00h
Type: Read-only
Default: 104Ch

4.3 Device ID Register

The device ID register contains a value assigned to the PCI1410A device by TI. The device identification for the PCI1410A device is AC50h.

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Device ID															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	1	0	1	0	1	1	0	0	0	1	0	1	0	0	0	0

Register: **Device ID**
Offset: 02h
Type: Read-only
Default: AC50h

4.4 Command Register

The command register provides control over the PCI1410A interface to the PCI bus. All bit functions adhere to the definitions in *PCI Local Bus Specification*. See Table 4–3 for the complete description of the register contents.

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Command															
Type	R	R	R	R	R	R	R	R/W	R	R/W	R/W	R	R	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **Command**
Offset: 04h
Type: Read-only, Read/Write
Default: 0000h

Table 4–3. Command Register Description

BIT	SIGNAL	TYPE	FUNCTION
15–10	RSVD	R	Reserved. Bits 15–10 return 0s when read.
9	FBB_EN	R	Fast back-to-back enable. The PCI1410A device does not generate fast back-to-back transactions; therefore, bit 9 returns 0 when read.
8	SERR_EN	R/W	System error ($\overline{SERR}$) enable. Bit 8 controls the enable for the $\overline{SERR}$ driver on the PCI interface. $\overline{SERR}$ can be asserted after detecting an address parity error on the PCI bus. Both bits 8 and 6 must be set for the PCI1410A device to report address parity errors. 0 = Disables the $\overline{SERR}$ output driver (default). 1 = Enables the $\overline{SERR}$ output driver.
7	STEP_EN	R	Address/data stepping control. The PCI1410A device does not support address/data stepping; therefore, bit 7 is hardwired to 0.
6	PERR_EN	R/W	Parity-error response enable. Bit 6 controls the PCI1410A device response to parity errors through $\overline{PERR}$. Data parity errors are indicated by asserting $\overline{PERR}$; address parity errors are indicated by asserting $\overline{SERR}$. 0 = PCI1410A device ignores detected parity error (default). 1 = PCI1410A device responds to detected parity errors.
5	VGA_EN	R/W	VGA palette snoop. When bit 5 is set to 1, palette snooping is enabled (that is, the PCI1410A device does not respond to palette register writes and snoops the data). When bit 5 is 0, the PCI1410A device treats all palette accesses like all other accesses.
4	MWI_EN	R	Memory write and invalidate enable. Bit 4 controls whether a PCI initiator device can generate memory write and invalidate commands. The PCI1410A controller does not support memory write and invalidate commands. It uses memory write commands instead; therefore, this bit is hardwired to 0.
3	SPECIAL	R	Special cycles. Bit 3 controls whether or not a PCI device ignores PCI special cycles. The PCI1410A device does not respond to special cycle operations; therefore, this bit is hardwired to 0.
2	MAST_EN	R/W	Bus-master control. Bit 2 controls whether or not the PCI1410A device can act as a PCI bus initiator (master). The PCI1410A device can take control of the PCI bus only when this bit is set. 0 = Disables the PCI1410A device's ability to generate PCI bus accesses (default). 1 = Enables the PCI1410A device's ability to generate PCI bus accesses.
1	MEM_EN	R/W	Memory space enable. Bit 1 controls whether or not the PCI1410A device can claim cycles in PCI memory space. 0 = Disables the PCI1410A device response to memory space accesses (default). 1 = Enables the PCI1410A device response to memory space accesses.
0	IO_EN	R/W	I/O space control. Bit 0 controls whether or not the PCI1410A device can claim cycles in PCI I/O space. 0 = Disables the PCI1410A device response to I/O space accesses (default). 1 = Enables the PCI1410A device response to I/O space accesses.

4.5 Status Register

The status register provides device information to the host system. Bits in this register can be read normally. A bit in the status register is reset when a 1 is written to that bit location; a 0 written to a bit location has no effect. All bit functions adhere to the definitions in the *PCI Local Bus Specification*. PCI bus status is shown through each function. See Table 4–4 for a complete description of the register contents.

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Status															
Type	R/C	R/C	R/C	R/C	R/C	R	R	R/C	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0

Register: **Status**
Offset: 06h
Type: Read-only, Read/Clear
Default: 0210h

Table 4–4. Status Register Description

BIT	SIGNAL	TYPE	FUNCTION
15	PAR_ERR	R/C	Detected parity error. Bit 15 is set when a parity error is detected (either address or data). Write a 1 to clear this bit.
14	SYS_ERR	R/C	Signaled system error. Bit 14 is set when $\overline{\text{SERR}}$ is enabled and the PCI1410A device signals a system error to the host. Write a 1 to clear this bit.
13	MABORT	R/C	Received master abort. Bit 13 is set when a cycle initiated by the PCI1410A device on the PCI bus is terminated by a master abort. Write a 1 to clear this bit.
12	TABT_REC	R/C	Received target abort. Bit 12 is set when a cycle initiated by the PCI1410A device on the PCI bus is terminated by a target abort. Write a 1 to clear this bit.
11	TABT_SIG	R/C	Signaled target abort. Bit 11 is set by the PCI1410A device when it terminates a transaction on the PCI bus with a target abort. Write a 1 to clear this bit.
10–9	PCI_SPEED	R	$\overline{\text{DEVSEL}}$ timing. These bits encode the timing of $\overline{\text{DEVSEL}}$ and are hardwired 01b, indicating that the PCI1410A device asserts PCI_SPEED at a medium speed on nonconfiguration cycle accesses.
8	DATAPAR	R/C	Data parity error detected. Write a 1 to clear this bit. 0 = The conditions for setting bit 8 have not been met. 1 = A data parity error occurred, and the following conditions were met: a. $\overline{\text{PERR}}$ was asserted by any PCI device, including the PCI1410A device. b. The PCI1410A device was the bus master during the data parity error. c. Bit 6 (PERR_EN) in the command register (PCI offset 04h, see Section 4.4) is set.
7	FBB_CAP	R	Fast back-to-back capable. The PCI1410A device cannot accept fast back-to-back transactions; therefore, bit 7 is hardwired to 0.
6	UDF	R	User-definable feature support. The PCI1410A device does not support the user-definable features; therefore, bit 6 is hardwired to 0.
5	66MHZ	R	66-MHz capable. The PCI1410A device operates at a maximum PCLK frequency of 33 MHz; therefore, bit 5 is hardwired to 0.
4	CAPLIST	R	Capabilities list. Bit 4 returns 1 when read. This bit indicates that capabilities, in addition to standard PCI capabilities, are implemented. The linked list of PCI power management capabilities is implemented in this function.
3–0	RSVD	R	Reserved. Bits 3–0 return 0s when read.

4.6 Revision ID Register

The revision ID register indicates the silicon revision of the PCI1410A device.

Bit	7	6	5	4	3	2	1	0
Name	Revision ID							
Type	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	1	0

Register: **Revision ID**
Offset: 08h
Type: Read-only
Default: 02h

4.7 PCI Class Code Register

The class code register recognizes the PCI1410A device as a bridge device (06h) and CardBus bridge device (07h) with a 00h programming interface.

Bit	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	PCI class code																							
	Base class								Subclass								Programming interface							
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0

Register: **PCI class code**
Offset: 09h
Type: Read-only
Default: 06 0700h

4.8 Cache Line Size Register

The cache line size register is programmed by host software to indicate the system cache line size.

Bit	7	6	5	4	3	2	1	0
Name	Cache line size							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

Register: **Cache line size**
Offset: 0Ch
Type: Read/Write
Default: 00h

4.9 Latency Timer Register

The latency timer register specifies the latency timer for the PCI1410A device in units of PCI clock cycles. When the PCI1410A device is a PCI bus initiator and asserts $\overline{\text{FRAME}}$, the latency timer begins counting from zero. If the latency timer expires before the PCI1410A transaction has terminated, the PCI1410A device terminates the transaction when its $\overline{\text{GNT}}$ is deasserted.

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Latency timer							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

Register: **Latency timer**
Offset: 0Dh
Type: Read/Write
Default: 00h

4.10 Header Type Register

This register returns 02h when read, indicating that the PCI1410A configuration spaces adhere to the CardBus bridge PCI header. The CardBus bridge PCI header ranges from PCI register 0 to 7Fh, and 80h–FFh are user-definable extension registers.

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Header type							
Type	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	1	0

Register: **Header type**
Offset: 0Eh
Type: Read-only
Default: 02h

4.11 BIST Register

Because the PCI1410A device does not support a built-in self-test (BIST), this register returns the value of 00h when read.

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	BIST							
Type	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0

Register: **BIST**
Offset: 0Fh
Type: Read-only
Default: 00h

4.12 CardBus Socket/ExCA Base-Address Register

The CardBus socket/ExCA base-address register is programmed with a base address referencing the CardBus socket registers and the memory-mapped ExCA register set. Bits 31–12 are read/write and allow the base address to be located anywhere in the 32-bit PCI memory address space on a 4-Kbyte boundary. Bits 11–0 are read-only, returning 0s when read. When software writes all 1s to this register, the value read back is FFFF F000h, indicating that at least 4 Kbytes of memory address space are required. The CardBus registers start at offset 000h, and the memory-mapped ExCA registers begin at offset 800h.

Bit	**31**	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Name	CardBus socket/ExCA base-address															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	CardBus socket/ExCA base-address															
Type	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **CardBus socket/ExCA base-address**
Offset: 10h
Type: Read-only, Read/Write
Default: 0000 0000h

4.13 Capability Pointer Register

The capability pointer register provides a pointer into the PCI configuration header where the PCI power-management register block resides. PCI header doublewords at A0h and A4h provide the power-management (PM) registers. The socket has its own capability pointer register. This register returns A0h when read.

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Capability pointer							
Type	R	R	R	R	R	R	R	R
Default	1	0	1	0	0	0	0	0

Register: **Capability pointer**
Offset: 14h
Type: Read-only
Default: A0h

4.14 Secondary Status Register

The secondary status register is compatible with the PCI-to-PCI bridge secondary status register and indicates CardBus-related device information to the host system. This register is very similar to the status register (PCI offset 06h); status bits are cleared by writing a 1. See Table 4–5 for a complete description of the register contents.

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Secondary status															
Type	R/C	R/C	R/C	R/C	R/C	R	R	R/C	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

Register: **Secondary status**
Offset: 16h
Type: Read-only, Read/Clear
Default: 0200h

Table 4–5. Secondary Status Register Description

BIT	SIGNAL	TYPE	FUNCTION
15	CBPARITY	R/C	Detected parity error. Bit 15 is set when a CardBus parity error is detected (either address or data). Write a 1 to clear this bit.
14	CBSERR	R/C	Signaled system error. Bit 14 is set when $\overline{\text{CSERR}}$ is signaled by a CardBus card. The PCI1410A device does not assert $\overline{\text{CSERR}}$. Write a 1 to clear this bit.
13	CBMABORT	R/C	Received master abort. Bit 13 is set when a cycle initiated by the PCI1410A device on the CardBus bus is terminated by a master abort. Write a 1 to clear this bit.
12	REC_CBTA	R/C	Received target abort. Bit 12 is set when a cycle initiated by the PCI1410A device on the CardBus bus is terminated by a target abort. Write a 1 to clear this bit.
11	SIG_CBTA	R/C	Signaled target abort. Bit 11 is set by the PCI1410A device when it terminates a transaction on the CardBus bus with a target abort. Write a 1 to clear this bit.
10–9	CB_SPEED	R	$\overline{\text{CDEVSEL}}$ timing. These bits encode the timing of $\overline{\text{CDEVSEL}}$ and are hardwired 01b, indicating that the PCI1410A device asserts CB_SPEED at a medium speed.
8	CB_DPAR	R/C	CardBus data parity error detected. Write a 1 to clear this bit. 0 = The conditions for setting bit 8 have not been met. 1 = A data parity error occurred and the following conditions were met: a. $\overline{\text{CPERR}}$ was asserted on the CardBus interface. b. The PCI1410A device was the bus master during the data parity error. c. Bit 0 (CPERREN) in the bridge control register (PCI offset 3Eh, see Section 4.25) is set.
7	CBFBB_CAP	R	Fast back-to-back capable. The PCI1410A device cannot accept fast back-to-back transactions; therefore, bit 7 is hardwired to 0.
6	CB_UDF	R	User-definable feature support. The PCI1410A device does not support the user-definable features; therefore, bit 6 is hardwired to 0.
5	CB66MHZ	R	66-MHz capable. The PCI1410A CardBus interface operates at a maximum CCLK frequency of 33 MHz; therefore, bit 5 is hardwired to 0.
4–0	RSVD	R	Reserved. Bits 4–0 return 0s when read.

4.15 PCI Bus Number Register

The PCI bus number register is programmed by the host system to indicate the bus number of the PCI bus to which the PCI1410A device is connected. The PCI1410A device uses this register in conjunction with the CardBus bus number (PCI offset 19h, see Section 4.16) and subordinate bus number (PCI offset 1Ah, see Section 4.17) registers to determine when to forward PCI configuration cycles to its secondary buses.

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	PCI bus number							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

Register: **PCI bus number**
Offset: 18h
Type: Read/Write
Default: 00h

4.16 CardBus Bus Number Register

The CardBus bus number register is programmed by the host system to indicate the bus number of the CardBus bus to which the PCI1410A device is connected. The PCI1410A device uses this register in conjunction with the PCI bus number (PCI offset 18h, see Section 4.15) and subordinate bus number (PCI offset 1Ah, see Section 4.17) registers to determine when to forward PCI configuration cycles to its secondary buses.

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	CardBus bus number							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

Register: **CardBus bus number**
Offset: 19h
Type: Read/Write
Default: 00h

4.17 Subordinate Bus Number Register

The subordinate bus number register is programmed by the host system to indicate the highest-numbered bus below the CardBus bus. The PCI1410A device uses this register in conjunction with the PCI bus number (PCI offset 18h, see Section 4.15) and CardBus bus number (PCI offset 19h, see Section 4.16) registers to determine when to forward PCI configuration cycles to its secondary buses.

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Subordinate bus number							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

Register: **Subordinate bus number**
Offset: 1Ah
Type: Read/Write
Default: 00h

4.18 CardBus Latency Timer Register

This register is programmed by the host system to specify the latency timer for the PCI1410A CardBus interface in units of CCLK cycles. When the PCI1410A device is a CardBus initiator and asserts $\overline{\text{CFRAME}}$, the CardBus latency timer begins counting. If the latency timer expires before the PCI1410A transaction has terminated, the PCI1410A device terminates the transaction at the end of the next data phase. A recommended minimum value for this register is 20h, which allows most transactions to be completed.

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	CardBus latency timer							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

Register: **CardBus latency timer**
Offset: 1Bh
Type: Read/Write
Default: 00h

4.19 Memory Base Registers 0, 1

The memory base registers indicate the lower address of a PCI memory address range. These registers are used by the PCI1410A device to determine when to forward a memory transaction to the CardBus bus and when to forward a CardBus cycle to the PCI bus. Bits 31–12 of these registers are read/write and allow the memory base to be located anywhere in the 32-bit PCI memory space on 4-Kbyte boundaries. Bits 11–0 are read-only and always return 0s. Write transactions to these bits have no effect. Bits 8 (PREFETCH0) and 9 (PREFETCH1) of the bridge control register (PCI offset 3Eh, see Section 4.25) specify whether memory windows 0 and 1 are prefetchable or nonprefetchable. The memory base register or the memory limit register must be nonzero for the PCI1410A device to claim any memory transactions through the CardBus memory windows (that is, these windows are not enabled by default to pass the first 4 Kbytes of memory to CardBus).

Bit	**31**	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Name	Memory base registers 0, 1															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Memory base registers 0, 1															
Type	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **Memory base register 0**
Offset: 1Ch
Register: **Memory base register 1**
Offset: 24h
Type: Read-only, Read/Write
Default: 0000 0000h

4.20 Memory Limit Registers 0, 1

The memory limit registers indicate the upper address of a PCI memory address range. These registers are used by the PCI1410A device to determine when to forward a memory transaction to the CardBus bus and when to forward a CardBus cycle to the PCI bus. Bits 31–12 of these registers are read/write and allow the memory base to be located anywhere in the 32-bit PCI memory space on 4-Kbyte boundaries. Bits 11–0 are read-only and always return 0s. Write transactions to these bits have no effect. Bits 8 (PREFETCH0) and 9 (PREFETCH1) of the bridge control register (PCI offset 3Eh, see Section 4.25) specify whether memory windows 0 and 1 are prefetchable or nonprefetchable. The memory base register or the memory limit register must be nonzero for the PCI1410A device to claim any memory transactions through the CardBus memory windows (that is, these windows are not enabled by default to pass the first 4 Kbytes of memory to CardBus).

Bit	**31**	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Name	Memory limit registers 0, 1															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Memory limit registers 0, 1															
Type	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **Memory limit registers 0, 1**
Offset: 20h, 28h
Type: Read-only, Read/Write
Default: 0000 0000h

4.21 I/O Base Registers 0, 1

The I/O base registers indicate the lower address of a PCI I/O address range. These registers are used by the PCI1410A device to determine when to forward an I/O transaction to the CardBus bus and when to forward a CardBus cycle to the PCI bus. The lower 16 bits of these registers locate the bottom of the I/O window within a 64-Kbyte page, and the upper 16 bits (31–16) are a page register that locates this 64-Kbyte page in 32-bit PCI I/O address space. Bits 31–2 are read/write. Bits 1 and 0 are read-only and always return 0s, forcing the I/O window to be aligned on a natural doubleword boundary.

> **NOTE:** Either the I/O base or the I/O limit register must be nonzero to enable any I/O transactions.

Bit	**31**	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Name	I/O base registers 0, 1															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	I/O base registers 0, 1															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **I/O base registers 0, 1**
Offset: 2Ch, 34h
Type: Read-only, Read/Write
Default: 0000 0000h

4.22 I/O Limit Registers 0, 1

The I/O limit registers indicate the upper address of a PCI I/O address range. These registers are used by the PCI1410A device to determine when to forward an I/O transaction to the CardBus bus and when to forward a CardBus cycle to the PCI bus. The lower 16 bits of these registers locate the top of the I/O window within a 64-Kbyte page, and the upper 16 bits are a page register that locates this 64-Kbyte page in 32-bit PCI I/O address space. Bits 15–2 are read/write and allow the I/O limit address to be located anywhere in the 64-Kbyte page (indicated by bits 31–16 of the appropriate I/O base register) on doubleword boundaries.

Bits 31–16 are read-only and always return 0s when read. The page is set in the I/O base register. Bits 1 and 0 are read-only and always return 0s, forcing I/O windows to be aligned on a natural doubleword boundary. Write transactions to read-only bits have no effect. The PCI1410A device assumes that the lower 2 bits of the limit address are 1s.

NOTE: The I/O base or the I/O limit register must be nonzero to enable an I/O transaction.

Bit	**31**	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Name	I/O limit registers 0, 1															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	I/O limit registers 0, 1															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **I/O limit registers 0, 1**
Offset: 30h, 38h
Type: Read-only, Read/Write
Default: 0000 0000h

4.23 Interrupt Line Register

The interrupt line register communicates interrupt line routing information.

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Interrupt line							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	1	1	1	1	1	1	1	1

Register: **Interrupt line**
Offset: 3Ch
Type: Read/Write
Default: FFh

4.24 Interrupt Pin Register

The value read from the interrupt pin register is function dependent and depends on the interrupt signaling mode, selected through bits 2–1 (INTMODE field) of the device control register (PCI offset 92h, see Section 4.33). The PCI1410A device defaults to serialized PCI and ISA interrupt mode.

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Interrupt pin							
Type	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	1

Register: **Interrupt pin**
Offset: 3Dh
Type: Read-only
Default: 01h

4.25 Bridge Control Register

The bridge control register provides control over various PCI1410A bridging functions. See Table 4–6 for a complete description of the register contents.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Bridge control															
Type	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0

Register: **Bridge control**
Offset: 3Eh
Type: Read-only, Read/Write
Default: 0340h

Table 4–6. Bridge Control Register Description

BIT	SIGNAL	TYPE	FUNCTION
15–11	RSVD	R	Reserved. Bits 15–11 return 0s when read.
10	POSTEN	R/W	Write posting enable. Enables write posting to and from the CardBus sockets. Write posting enables posting of write data on burst cycles. Operating with write posting disabled inhibits performance on burst cycles. Note that burst write data can be posted, but various write transactions cannot.
9	PREFETCH1	R/W	Memory window 1 type. Bit 9 specifies whether or not memory window 1 is prefetchable. This bit is socket dependent. Bit 9 is encoded as: 0 = Memory window 1 is nonprefetchable. 1 = Memory window 1 is prefetchable (default).
8	PREFETCH0	R/W	Memory window 0 type. Bit 8 specifies whether or not memory window 0 is prefetchable. This bit is encoded as: 0 = Memory window 0 is nonprefetchable. 1 = Memory window 0 is prefetchable (default).
7	INTR	R/W	PCI interrupt – IREQ routing enable. Bit 7 selects whether PC Card functional interrupts are routed to PCI interrupts or to the IRQ specified in the ExCA registers. 0 = Functional interrupts are routed to PCI interrupts (default). 1 = Functional interrupts are routed by ExCAs.
6	CRST	R/W	CardBus reset. When bit 6 is set, $\overline{\text{CRST}}$ is asserted on the CardBus interface. $\overline{\text{CRST}}$ also can be asserted by passing a $\overline{\text{PRST}}$ assertion to CardBus. 0 = $\overline{\text{CRST}}$ is deasserted. 1 = $\overline{\text{CRST}}$ is asserted (default).
5	MABTMODE	R/W	Master abort mode. Bit 5 controls how the PCI1410A device responds to a master abort when the PCI1410A device is an initiator on the CardBus interface. 0 = Master aborts are not reported (default). 1 = Signal target abort on PCI and $\overline{\text{SERR}}$ (if enabled)
4	RSVD	R	Reserved. Bit 4 returns 0 when read.
3	VGAEN	R/W	VGA enable. Bit 3 affects how the PCI1410A device responds to VGA addresses. When this bit is set, accesses to VGA addresses are forwarded.
2	ISAEN	R/W	ISA mode enable. Bit 2 affects how the PCI1410A device passes I/O cycles within the 64-Kbyte ISA range. This bit is not common between sockets. When this bit is set, the PCI1410A device does not forward the last 768 bytes of each 1K I/O range to CardBus.
1	CSERREN	R/W	$\overline{\text{CSERR}}$ enable. Bit 1 controls the response of the PCI1410A device to $\overline{\text{CSERR}}$ signals on the CardBus bus. This bit is common between the two sockets. 0 = $\overline{\text{CSERR}}$ is not forwarded to PCI $\overline{\text{SERR}}$. 1 = $\overline{\text{CSERR}}$ is forwarded to PCI $\overline{\text{SERR}}$.
0	CPERREN	R/W	CardBus parity error response enable. Bit 0 controls the response of the PCI1410A device to CardBus parity errors. This bit is common between the two sockets. 0 = CardBus parity errors are ignored. 1 = CardBus parity errors are reported using $\overline{\text{CPERR}}$.

4.26 Subsystem Vendor ID Register

The subsystem vendor ID register is used for system and option-card identification purposes and may be required for certain operating systems. This register is read-only or read/write, depending on the setting of bit 5 (SUBSYSRW) in the system control register (PCI offset 80h, see Section 4.29).

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Subsystem vendor ID															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **Subsystem vendor ID**
Offset: 40h
Type: Read-only (Read/Write if enabled by SUBSYSRW)
Default: 0000h

4.27 Subsystem ID Register

The subsystem ID register is used for system and option-card identification purposes and may be required for certain operating systems. This register is read-only or read/write, depending on the setting of bit 5 (SUBSYSRW) in the system control register (PCI offset 80h, see Section 4.29).

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Subsystem ID															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **Subsystem ID**
Offset: 42h
Type: Read-only (Read/Write if enabled by SUBSYSRW)
Default: 0000h

4.28 PC Card 16-Bit I/F Legacy-Mode Base-Address Register

The PCI1410A device supports the index/data scheme of accessing the ExCA registers, which is mapped by this register. An address written to this register is the address for the index register and the address + 1 is the data address. Using this access method, applications requiring index/data ExCA access can be supported. The base address can be mapped anywhere in 32-bit I/O space on a word boundary; hence, bit 0 is read-only, returning 1 when read. See Section 5, *ExCA Compatibility Registers*, for register offsets.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	PC Card 16-bit I/F legacy-mode base address															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	PC Card 16-bit I/F legacy-mode base address															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Register: **PC Card 16-bit I/F legacy-mode base address**
Offset: 44h
Type: Read-only, Read/Write
Default: 0000 0001h

4.29 System Control Register

System-level initializations are performed through programming this doubleword register. See Table 4–7 for a complete description of the register contents.

Bit	**31**	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Name	System control															
Type	R/W	R/W	R	R	R/W	R/W	R/C	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	System control															
Type	R/W	R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	1	0	0	1	0	0	0	0	0	1	1	0	0	0	0	0

Register: **System control**
Type: Read-only, Read/Write, Read/Clear
Offset: 80h
Default: 0044 9060h

Table 4–7. System Control Register Description

BIT	SIGNAL	TYPE	FUNCTION
31–30	SER_STEP	R/W	Serialized PCI interrupt routing step. Bits 31 and 30 configure the serialized PCI interrupt stream signaling and accomplish an even distribution of interrupts signaled on the four PCI interrupt slots. Bits 31 and 30 are encoded as follows: 00 = $\overline{\text{INTA}}$ signaled in $\overline{\text{INTA}}$ IRQSER slots 01 = $\overline{\text{INTA}}$ signaled in $\overline{\text{INTB}}$ IRQSER slots 10 = $\overline{\text{INTA}}$ signaled in $\overline{\text{INTC}}$ IRQSER slots 11 = $\overline{\text{INTA}}$ signaled in $\overline{\text{INTD}}$ IRQSER slots
29–28	RSVD	R	Reserved. Bits 29 and 28 return 0s when read.
27	OSEN	R/W	Internal oscillator enable. 0 = Internal oscillator is disabled (default). 1 = Internal oscillator is enabled.
26	SMIROUTE	R/W	SMI interrupt routing. Bit 26 selects whether IRQ2 or CSC is signaled when a write occurs to power a PC Card socket. 0 = PC Card power change interrupts are routed to IRQ2 (default). 1 = A CSC interrupt is generated on PC Card power changes.
25	SMISTATUS	R/C	SMI interrupt status. This bit is set when bit 24 (SMIENB) is set and a write occurs to set the socket power. Writing a 1 to bit 25 clears the status. 0 = SMI interrupt is signaled (default). 1 = SMI interrupt is not signaled.
24	SMIENB	R/W	SMI interrupt mode enable. When bit 24 is set and a write to the socket power control occurs, the SMI interrupt signaling is enabled and generates an interrupt. This bit is shared and defaults to 0 (disabled).
23	RSVD	R	Reserved. Bit 23 returns 0 when read.
22	CBRSVD	R/W	CardBus reserved-terminals signaling. When a CardBus card is inserted and bit 22 is set, the RSVD CardBus terminals are driven low. When this bit is 0, these signals are placed in a high-impedance state. 0 = Place CardBus RSVD in a high-impedance state 1 = Drive Cardbus RSVD low (default)
21	VCCPROT	R/W	V_{CC} protection enable. 0 = V_{CC} protection is enabled for 16-bit cards (default). 1 = V_{CC} protection is disabled for 16-bit cards.
20	REDUCEZV	R/W	Reduced zoomed-video enable. When this bit is enabled, pins ADDR25–ADDR22 of the card interface for PC Card-16 cards are placed in the high-impedance state. This bit should not be set for normal ZV operation. This bit is encoded as: 0 = Reduced ZV is disabled (default). 1 = Reduced ZV is enabled.
19	CDREQEN	R/W	PC/PCI DMA card enable. When bit 19 is set, the PCI1410A device allows 16-bit PC Cards to request PC/PCI DMA using the $\overline{\text{DREQ}}$ signaling. $\overline{\text{DREQ}}$ is selected through the socket DMA register (PCI offset 94h, see Section 4.35). 0 = Ignore $\overline{\text{DREQ}}$ signaling from PC Cards (default) 1 = Signal DMA request on $\overline{\text{DREQ}}$
18–16	CDMACHAN	R/W	PC/PCI DMA channel assignment. Bits 18–16 are encoded as: 0–3 = 8-bit DMA channels 4 = Not used (default) 5–7 = 16-bit DMA channels
15	MRBURSTDN	R/W	Memory-read burst-enable downstream. When bit 15 is set, memory-read transactions are allowed to burst downstream. 0 = Downstream memory-read burst is disabled. 1 = Downstream memory-read burst is enabled (default).
14	MRBURSTUP	R/W	Memory-read burst-enable upstream. When bit 14 is set, the PCI1410A device allows memory-read transactions to burst upstream. 0 = Upstream memory-read burst is disabled (default). 1 = Upstream memory-read burst is enabled.

Table 4–7. System Control Register Description (Continued)

BIT	SIGNAL	TYPE	FUNCTION
13	SOCACTIVE	R	Socket activity status. When set, bit 13 indicates access has been performed to or from a PC card and is cleared upon read of this status bit. 0 = No socket activity (default) 1 = Socket activity
12	RSVD	R	Reserved. Bit 12 returns 1 when read.
11	PWRSTREAM	R	Power stream in progress status bit. When set, bit 11 indicates that a power stream to the power switch is in progress and a powering change has been requested. This bit is cleared when the power stream is complete. 0 = Power stream is complete and delay has expired. 1 = Power stream is in progress.
10	DELAYUP	R	Power-up delay in progress status. When set, bit 9 indicates that a power-up stream has been sent to the power switch and proper power may not yet be stable. This bit is cleared when the power-up delay has expired.
9	DELAYDOWN	R	Power-down delay in progress status. When set, bit 10 indicates that a power-down stream has been sent to the power switch and proper power may not yet be stable. This bit is cleared when the power-down delay has expired.
8	INTERROGATE	R	Interrogation in progress. When set, bit 8 indicates an interrogation is in progress and clears when interrogation completes. This bit is socket dependent. 0 = Interrogation is not in progress (default). 1 = Interrogation is in progress.
7	AUTOPWRSWEN	R/W	Auto power-switch enable. 0 = Bit 5 (AUTOPWRSWEN) in ExCA power control register (ExCA offset 02h, see Section 5.3) is disabled (default). 1 = Bit 5 (AUTOPWRSWEN) in ExCA power control register (ExCA offset 02h, see Section 5.3) is enabled.
6	PWRSAVINGS	R/W	Power savings mode enable. When this bit is set, if a CB card is inserted, idle, and without a CB clock, the applicable CB state machine is not clocked.
5	SUBSYSRW	R/W	Subsystem ID (PCI offset 42h, see Section 4.27), subsystem vendor ID (PCI offset 40h, see Section 4.26), ExCA identification and revision (ExCA offset 00h, see Section 5.1) registers read/write enable. 0 = Subsystem ID, subsystem vendor ID, ExCA identification and revision registers are read/write. 1 = Subsystem ID, subsystem vendor ID, ExCA identification and revision registers are read-only (default).
4	CB_DPAR	R/W	CardBus data parity SERR signaling enable. 0 = CardBus data parity is not signaled on PCI SERR. 1 = CardBus data parity is signaled on PCI SERR.
3	CDMA_EN	R/W	PC/PCI DMA enable. Bit 3 enables PC/PCI DMA when set if MFUNC0–MFUNC6 are configured for centralized DMA. 0 = Centralized DMA is disabled (default). 1 = Centralized DMA is enabled.
2	ExCAPower	R/W	ExCA power control bit. Enabled by selecting the 82365SL mode. 0 = Enables 3.3 V. 1 = Enables 5 V.
1	KEEPCLK	R/W	Keep clock. This bit works with PCI and CB CLKRUN protocols. 0 = Allows normal functioning of both CLKRUN protocols (default). 1 = Does not allow CB clock or PCI clock to be stopped using the CLKRUN protocols.
0	RIMUX	R/W	RI_OUT/PME multiplex enable. 0 = RI_OUT and PME are both routed to the RI_OUT/PME terminal. If both are enabled at the same time, RI_OUT has precedence over PME. 1 = Only PME is routed to the RI_OUT/PME terminal.

4.30 Multifunction Routing Register

The multifunction routing register is used to configure the MFUNC0–MFUNC6 terminals. These terminals can be configured for various functions. All multifunction terminals default to the general-purpose input configuration. This register is intended to be programmed once at power-on initialization. The default value for this register also can be loaded through a serial bus EEPROM. See Table 4–8 for a complete description of the register contents.

Bit	**31**	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Name	Multifunction routing															
Type	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Multifunction routing															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **Multifunction routing**
Offset: 8Ch
Type: Read-only, Read/Write
Default: 0000 0000h

Table 4–8. Multifunction Routing Register Description

BIT	SIGNAL	TYPE	FUNCTION
31–28	RSVD	R	Bits 31–28 return 0s when read.
27–24	MFUNC6	R/W	Multifunction terminal 6 configuration. These bits control the internal signal mapped to the MFUNC6 terminal as follows: 0000 = RSVD 0100 = IRQ4 1000 = IRQ8 1100 = IRQ12 0001 = CLKRUN 0101 = IRQ5 1001 = IRQ9 1101 = IRQ13 0010 = IRQ2 0110 = IRQ6 1010 = IRQ10 1110 = IRQ14 0011 = IRQ3 0111 = IRQ7 1011 = IRQ11 1111 = IRQ15
23–20	MFUNC5	R/W	Multifunction terminal 5 configuration. These bits control the internal signal mapped to the MFUNC5 terminal as follows: 0000 = GPI4 0100 = IRQ4 1000 = CAUDPWM 1100 = LED_SKT 0001 = GPO4 0101 = RSVD 1001 = IRQ9 1101 = LED_SKT 0010 = PCGNT 0110 = ZVSTAT 1010 = IRQ10 1110 = GPE 0011 = IRQ3 0111 = ZVSEL0 1011 = IRQ11 1111 = IRQ15
19–16	MFUNC4	R/W	Multifunction terminal 4 configuration. These bits control the internal signal mapped to the MFUNC4 terminal as follows: NOTE: When the serial bus mode is implemented by pulling up the VCCD0 and VCCD1 terminals, the MFUNC4 terminal provides the SCL signaling. 0000 = GPI3 0100 = IRQ4 1000 = CAUDPWM 1100 = RI_OUT 0001 = GPO3 0101 = IRQ5 1001 = IRQ9 1101 = LED_SKT 0010 = LOCK PCI 0110 = ZVSTAT 1010 = IRQ10 1110 = GPE 0011 = IRQ3 0111 = ZVSEL0 1011 = IRQ11 1111 = IRQ15
15–12	MFUNC3	R/W	Multifunction terminal 3 configuration. These bits control the internal signal mapped to the MFUNC3 terminal as follows: 0000 = RSVD 0100 = IRQ4 1000 = IRQ8 1100 = IRQ12 0001 = IRQSER 0101 = IRQ5 1001 = IRQ9 1101 = IRQ13 0010 = IRQ2 0110 = IRQ6 1010 = IRQ10 1110 = IRQ14 0011 = IRQ3 0111 = IRQ7 1011 = IRQ11 1111 = IRQ15
11–8	MFUNC2	R/W	Multifunction terminal 2 configuration. These bits control the internal signal mapped to the MFUNC2 terminal as follows: 0000 = GPI2 0100 = IRQ4 1000 = CAUDPWM 1100 = RI_OUT 0001 = GPO2 0101 = IRQ5 1001 = IRQ9 1101 = RSVD 0010 = PCREQ 0110 = ZVSTAT 1010 = IRQ10 1110 = GPE 0011 = IRQ3 0111 = ZVSEL0 1011 = IRQ11 1111 = IRQ7

Table 4–8. Multifunction Routing Register Description (Continued)

BIT	SIGNAL	TYPE	FUNCTION
7–4	MFUNC1	R/W	Multifunction terminal 1 configuration. These bits control the internal signal mapped to the MFUNC1 terminal as follows: NOTE: When the serial bus mode is implemented by pulling up the VCCD0 and VCCD1 terminals, the MFUNC1 terminal provides the SDA signaling. 0000 = GPI1 0100 = IRQ4 1000 = CAUDPWM 1100 = LED_SKT 0001 = GPO1 0101 = IRQ5 1001 = IRQ9 1101 = IRQ13 0010 = RSVD 0110 = ZVSTAT 1010 = IRQ10 1110 = GPE 0011 = IRQ3 0111 = ZVSEL0 1011 = IRQ11 1111 = IRQ15
3–0	MFUNC0	R/W	Multifunction terminal 0 configuration. These bits control the internal signal mapped to the MFUNC0 terminal as follows: 0000 = GPI0 0100 = IRQ4 1000 = CAUDPWM 1100 = LED_SKT 0001 = GPO0 0101 = IRQ5 1001 = IRQ9 1101 = IRQ13 0010 = INTA 0110 = ZVSTAT 1010 = IRQ10 1110 = GPE 0011 = IRQ3 0111 = ZVSEL0 1011 = IRQ11 1111 = IRQ15

4.31 Retry Status Register

The retry status register enables the retry timeout counters and displays the retry expiration status. The flags are set when the PCI1410A device retries a PCI or CardBus master request and the master does not return within 2^{15} PCI clock cycles. The flags are cleared by writing a 1 to the bit. These bits are expected to be incorporated into the PCI command, PCI status, and bridge control registers by the PCI SIG. See Table 4–9 for a complete description of the register contents.

Bit	7	6	5	4	3	2	1	0
Name	Retry status							
Type	R/W	R/W	R	R	R/C	R	R/C	R
Default	1	1	0	0	0	0	0	0

Register: **Retry status**
Offset: 90h
Type: Read-only, Read/Write, Read/Clear
Default: C0h

Table 4–9. Retry Status Register Description

BIT	SIGNAL	TYPE	FUNCTION
7	PCIRETRY	R/W	PCI retry timeout counter enable. Bit 7 is encoded: 0 = PCI retry counter is disabled. 1 = PCI retry counter is enabled (default).
6	CBRETRY	R/W	CardBus retry timeout counter enable. Bit 6 is encoded: 0 = CardBus retry counter is disabled. 1 = CardBus retry counter is enabled (default).
5–4	RSVD	R	Reserved. Bits 5 and 4 return 0s when read.
3	TEXP_CB	R/C	CardBus target retry expired. Write a 1 to clear bit 3. 0 = Inactive (default) 1 = Retry has expired.
2	RSVD	R	Reserved. Bit 2 returns 0 when read.
1	TEXP_PCI	R/C	PCI target retry expired. Write a 1 to clear bit 1. 0 = Inactive (default) 1 = Retry has expired.
0	RSVD	R	Reserved. Bit 0 returns 0 when read.

4.32 Card Control Register

The card control register is provided for PCI1130 compatibility. RI_OUT is enabled through this register. See Table 4–10 for a complete description of the register contents.

Bit	7	6	5	4	3	2	1	0
Name	Card control							
Type	R/W	R/W	R/W	R	R	R/W	R/W	R/C
Default	0	0	0	0	0	0	0	0

Register: **Card control**
Offset: 91h
Type: Read-only, Read/Write, Read/Clear
Default: 00h

Table 4–10. Card Control Register Description

BIT	SIGNAL	TYPE	FUNCTION
7	RIENB	R/W	Ring indicate output enable. 0 = Disables any routing of RI_OUT signal (default). 1 = Enables RI_OUT signal for routing to the RI_OUT/PME terminal, when bit 0 (RIMUX) in the system control register (PCI offset 80h, see Section 4.29) is set to 0, and for routing to MFUNC2 or MFUNC4.
6	ZVENABLE	R/W	Compatibility ZV mode enable. When set, the PC Card socket interface ZV terminals enter a high-impedance state. This bit defaults to 0.
5	No function	R/W	This bit has no assigned function.
4–3	RSVD	R	Reserved. Bits 4 and 3 return 0 when read.
2	AUD2MUX	R/W	CardBus audio-to-IRQMUX. When set, the CAUDIO CardBus signal is routed to the corresponding multifunction terminal, which may be configured for CAUDPWM.
1	SPKROUTEN	R/W	Speaker out enable. When bit 1 is set, SPKR on the PC Card is enabled and is routed to SPKROUT. The SPKROUT terminal drives data only when the socket's SPKROUTEN bit is set. This bit is encoded as: 0 = SPKR to SPKROUT is not enabled (default). 1 = SPKR to SPKROUT is enabled.
0	IFG	R/C	Interrupt flag. Bit 0 is the interrupt flag for 16-bit I/O PC Cards and for CardBus cards. Bit 0 is set when a functional interrupt is signaled from a PC Card interface. Write back a 1 to clear this bit. 0 = No PC Card functional interrupt is detected (default). 1 = PC Card functional interrupt is detected.

4.33 Device Control Register

The device control register is provided for PCI1130 compatibility. The interrupt mode select and the socket-capable force bits are programmed through this register. See Table 4–11 for a complete description of the register contents.

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Device control							
Type	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W
Default	0	1	1	0	0	1	1	0

Register: **Device control**
Offset: 92h
Type: Read-only, Read/Write
Default: 66h

Table 4–11. Device Control Register Description

BIT	SIGNAL	TYPE	FUNCTION
7	SKTPWR_LOCK	R/W	Socket-power lock bit. When this bit is set to 1, software cannot power down the PC Card socket while in D3. This may be necessary to support wake-up on LAN or RING if the operating system is programmed to power down a socket when the CardBus controller is placed in the D3 state.
6	3VCAPABLE	R/W	3-V socket-capable force 0 = Not 3-V capable 1 = 3-V capable (default)
5	IO16V2	R/W	Diagnostic bit. This bit defaults to 1.
4	RSVD	R	Reserved. Bit 4 returns 0 when read.
3	TEST	R/W	TI test. Only a 0 should be written to bit 3.
2–1	INTMODE	R/W	Interrupt signaling mode. Bits 2 and 1 select the interrupt signaling mode. The interrupt signaling mode bits are encoded: 00 = Parallel PCI interrupts only 01 = Parallel IRQ and parallel PCI interrupts 10 = IRQ serialized interrupts and parallel PCI interrupt 11 = IRQ and PCI serialized interrupts (default)
0	RSVD	R/W	Reserved. Bit 0 is reserved for test purposes. Only 0 should be written to this bit.

4.34 Diagnostic Register

The diagnostic register is provided for internal TI test purposes. It is a read/write register, but only 0s should be written to it. See Table 4–12 for a complete description of the register contents.

Bit	7	6	5	4	3	2	1	0
Name	Diagnostic							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	1	0	0	0	0	1

Register: **Diagnostic**
Offset: 93h
Type: Read/Write
Default: 21h

Table 4–12. Diagnostic Register Description

BIT	SIGNAL	TYPE	FUNCTION
7	TRUE_VAL	R/W	This bit defaults to 0. This bit is encoded as: 0 = Reads true values in PCI vendor ID and PCI device ID registers (default). 1 = Reads all 1s in the PCI vendor ID and PCI device ID registers.
6	RSVD	R/W	Reserved. Bit 6 returns 0 when read.
5	CSC	R/W	CSC interrupt routing control 0 = CSC interrupts are routed to PCI if ExCA 803 (see Section 5.4) bit 4 = 1. 1 = CSC interrupts are routed to PCI if ExCA 805 (see Section 5.6) bits 7–4 = 0000b (default). In this case, the setting of ExCA 803 bit 4 is a don't care.
4	DIAG4	R/W	Diagnostic RETRY_DIS. Delayed transaction disable.
3	DIAG3	R/W	Diagnostic RETRY_EXT. Extends the latency from 16 to 64.
2	DIAG2	R/W	Diagnostic DISCARD_TIM_SEL_CB. Set = 2^{10}, reset = 2^{15}.
1	DIAG1	R/W	Diagnostic DISCARD_TIM_SEL_PCI. Set = 2^{10}, reset = 2^{15}.
0	ASYNCINT	R/W	Asynchronous interrupt enable. 0 = CSC interrupt is not generated asynchronously. 1 = CSC interrupt is generated asynchronously (default).

4.35 Socket DMA Register 0

The socket DMA register 0 provides control over the PC Card DMA request (DREQ) signaling. See Table 4–13 for a complete description of the register contents.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Socket DMA register 0															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Socket DMA register 0															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **Socket DMA register 0**
Offset: 94h
Type: Read-only, Read/Write
Default: 0000 0000h

Table 4–13. Socket DMA Register 0 Description

BIT	SIGNAL	TYPE	FUNCTION
31–2	RSVD	R	Reserved. Bits 31–2 return 0s when read.
1–0	DREQPIN	R/W	DMA request (DREQ). Bits 1 and 0 indicate which pin on the 16-bit PC Card interface acts as DREQ during DMA transfers. This field is encoded as: 00 = Socket not configured for DMA (default). 01 = DREQ uses SPKR. 10 = DREQ uses IOIS16. 11 = DREQ uses INPACK.

4.36 Socket DMA Register 1

The socket DMA register 1 provides control over the distributed DMA (DDMA) registers and the PCI portion of DMA transfers. The DMA base address locates the DMA registers in a 16-byte region within the first 64 Kbytes of PCI I/O address space. See Table 4–14 for a complete description of the register contents.

NOTE: 32-bit transfers are not supported; the maximum transfer possible for 16-bit PC Cards is 16 bits.

Bit	**31**	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Name	Socket DMA register 1															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Socket DMA register 1															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **Socket DMA register 1**
Offset: 98h
Type: Read-only, Read/Write
Default: 0000 0000h

Table 4–14. Socket DMA Register 1 Description

BIT	SIGNAL	TYPE	FUNCTION
31–16	RSVD	R	Reserved. Bits 31–16 return 0s when read.
15–4	DMABASE	R/W	DMA base address. Locates the socket's DMA registers in PCI I/O space. This field represents a 16-bit PCI I/O address. The upper 16 bits of the address are hardwired to 0, forcing this window to within the lower 64 Kbytes of I/O address space. The lower 4 bits are hardwired to 0 and are included in the address decode. Thus, the window is aligned to a natural 16-byte boundary.
3	EXTMODE	R	Extended addressing. This feature is not supported by the PCI4410 and always returns a 0.
2–1	XFERSIZE	R/W	Transfer size. Bits 2 and 1 specify the width of the DMA transfer on the PC Card interface and are encoded as: 00 = Transfers are 8 bits (default). 01 = Transfers are 16 bits. 10 = Reserved 11 = Reserved
0	DDMAEN	R/W	DDMA registers decode enable. Enables the decoding of the distributed DMA registers based on the value of bits 15–4 (DMABASE field). 0 = Disabled (default) 1 = Enabled

4.37 Capability ID Register

The capability ID register identifies the linked list item as the register for PCI power management. The register returns 01h when read, which is the unique ID assigned by the PCI SIG for the PCI location of the capabilities pointer and the value.

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Capability ID							
Type	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	1

Register: **Capability ID**
Offset: A0h
Type: Read-only
Default: 01h

4.38 Next-Item Pointer Register

The next-item pointer register indicates the next item in the linked list of the PCI power-management capabilities. Because the PCI1410A functions include only one capabilities item, this register returns 0s when read.

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Next-item pointer							
Type	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0

Register: **Next-item pointer**
Offset: A1h
Type: Read-only
Default: 00h

4.39 Power Management Capabilities Register

This register contains information on the capabilities of the PC Card function related to power management. Both PCI1410A CardBus bridge functions support D0, D1, D2, and D3 power states. See Table 4–15 for a complete description of the register contents.

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Power management capabilities															
Type	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	1	1	1	1	1	1	1	0	0	0	1	0	0	0	0	1

Register: **Power management capabilities**
Offset: A2h
Type: Read/Write, Read-only
Default: FE21h

Table 4–15. Power Management Capabilities Register Description

BIT	SIGNAL	TYPE	FUNCTION
15	PME_Support	R/W	$\overline{\text{PME}}$ support. This 5-bit field indicates the power states from which the PCI1410A functions can assert $\overline{\text{PME}}$. A 0 (zero) for any bit indicates that the function cannot assert the $\overline{\text{PME}}$ signal while in that power state. These five bits return 11111b when read. Each of these bits is described below: Bit 15 defaults to the value 1, indicating the $\overline{\text{PME}}$ signal can be asserted from the $D3_{cold}$ state. This bit is R/W because wake-up support from $D3_{cold}$ is contingent on the system providing an auxiliary power source to the V_{CC} terminals. If the system designer chooses not to provide an auxiliary power source to the V_{CC} terminals for $D3_{cold}$ wake-up support, the BIOS should write a 0 to this bit.
14–11	PME_Support	R	Bit 14 contains the value 1, indicating that the $\overline{\text{PME}}$ signal can be asserted from $D3_{hot}$ state. Bit 13 contains the value 1, indicating that the $\overline{\text{PME}}$ signal can be asserted from D2 state. Bit 12 contains the value 1, indicating that the $\overline{\text{PME}}$ signal can be asserted from D1 state. Bit 11 contains the value 1, indicating that the $\overline{\text{PME}}$ signal can be asserted from the D0 state.
10	D2_Support	R	D2 support. Bit 10 returns a 1 when read, indicating that the CardBus function supports the D2 device power state.
9	D1_Support	R	D1 support. Bit 9 returns a 1 when read, indicating that the CardBus function supports the D1 device power state.
8–6	RSVD	R	Reserved. Bits 8–6 return 0s when read.
5	DSI	R	Device-specific initialization. Bit 5 returns 1 when read, indicating that the CardBus controller function requires special initialization (beyond the standard PCI configuration header) before the generic class device driver is able to use it.
4	AUX_PWR	R	Auxiliary power source. Bit 4 is meaningful only if bit 15 (PME_Support, $D3_{cold}$) is set. When bit 4 is set, it indicates that support for $\overline{\text{PME}}$ in $D3_{cold}$ requires auxiliary power supplied by the system by way of a proprietary delivery vehicle. When bit 4 is 0, it indicates that the function supplies its own auxiliary power source.
3	PMECLK	R	$\overline{\text{PME}}$ clock. Bit 3 returns 0 when read, indicating that no host bus clock is required for the PCI1410A device to generate $\overline{\text{PME}}$.
2–0	VERSION	R	Version. Bits 2–0 return 001b when read, indicating that there are four bytes of general-purpose power management (PM) registers as described in the *PCI Bus Power Management Interface Specification*.

4.40 Power Management Control/Status Register

The power management control/status register determines and changes the current power state of the PCI1410A CardBus function. The contents of this register are not affected by the internally-generated reset caused by the transition from $D3_{hot}$ to D0 state. All PCI, ExCA, and CardBus registers are reset as a result of a $D3_{hot}$ to D0 state transition. TI-specific registers, PCI power management registers, and the legacy base address register are not reset. See Table 4–16 for a complete description of the register contents.

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Power management control/status															
Type	R/C	R	R	R	R	R	R	R/W	R	R	R	R	R	R	R/W	R/W
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **Power management control/status**
Offset: A4h
Type: Read-only, Read/Write, Read/Write to Clear
Default: 0000h

Table 4–16. Power Management Control/Status Register Description

BIT	SIGNAL	TYPE	FUNCTION
15	PMESTAT	R/C	$\overline{\text{PME}}$ status. Bit 15 is set when the CardBus function normally would assert $\overline{\text{PME}}$, independent of the state of bit 8 (PME_EN). Bit 15 is cleared by a write back of 1, and this also clears $\overline{\text{PME}}$ if $\overline{\text{PME}}$ was asserted by this function. Writing a 0 to this bit has no effect.
14–13	DATASCALE	R	Data scale. This 2-bit field returns 0s when read. The CardBus function does not return any dynamic data, as indicated by bit 4 (DYN_DATA_PME_EN).
12–9	DATASEL	R	Data select. This 4-bit field returns 0s when read. The CardBus function does not return any dynamic data, as indicated by bit 4 (DYN_DATA_PME_EN).
8	PME_EN	R/W	$\overline{\text{PME}}$ enable. Bit 8 enables the function to assert $\overline{\text{PME}}$. If this bit is cleared, the assertion of $\overline{\text{PME}}$ is disabled.
7–5	RSVD	R	Reserved. Bits 7–5 return 0s when read.
4	DYN_DATA_PME_EN	R	Dynamic data $\overline{\text{PME}}$ enable. Bit 4 returns 0 when read, because the CardBus function does not report dynamic data.
3–2	RSVD	R	Reserved. Bits 3–2 return 0s when read.
1–0	PWR_STATE	R/W	Power state. This 2-bit field is used both to determine the current power state of a function and to set the function into a new power state. This field is encoded as: 00 = D0 01 = D1 10 = D2 11 = $D3_{hot}$

4.41 Power Management Control/Status Bridge Support Extensions Register

The power management control/status register bridge support extensions support PCI bridge-specific functionality. See Table 4–17 for a complete description of the register contents.

Bit	7	6	5	4	3	2	1	0
Name	Power management control/status bridge support extensions							
Type	R	R	R	R	R	R	R	R
Default	1	1	0	0	0	0	0	0

Register: **Power management control/status bridge support extensions**
Offset: A6h
Type: Read-only
Default: C0h

Table 4–17. Power Management Control/Status Bridge Support Extensions Register Description

BIT	SIGNAL	TYPE	FUNCTION
7	BPCC_EN	R	BPCC_Enable. Bus power/clock control enable. This bit returns 1 when read. This bit is encoded as: 0 = Bus power/clock control is disabled. 1 = Bus power/clock control is enabled (default). A 0 indicates that the bus power/clock control policies defined in the *PCI Bus Power Management Interface Specification* are disabled. When the bus power/clock control enable mechanism is disabled, the bridge power management control/status register power state field (PCI offset A4h, see Section 4.40, bits 1–0) cannot be used by the system software to control the power or the clock of the bridge secondary bus. A 1 indicates that the bus power/clock control mechanism is enabled.
6	B2_$\overline{\text{B3}}$	R	B2/B3 support for $D3_{hot}$. The state of this bit determines the action that is to occur as a direct result of programming the function to $D3_{hot}$. This bit is meaningful only if bit 7 (BPCC_EN) is a 1. This bit is encoded as: 0 = When the bridge is programmed to $D3_{hot}$, its secondary bus will have its power removed (B3). 1 = When the bridge function is programmed to $D3_{hot}$, its secondary bus PCI clock will be stopped (B2). (Default)
5–0	RSVD	R	Reserved. Bits 5–0 return 0s when read.

4.42 Power Management Data Register

The power management data register returns 0s when read, because the CardBus functions do not report dynamic data.

Bit	7	6	5	4	3	2	1	0
Name	Power management data							
Type	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0

Register: **Power management data**
Offset: A7h
Type: Read-only
Default: 00h

4.43 General-Purpose Event Status Register

The general-purpose event status register contains status bits that are set when events occur that are controlled by the general-purpose control register. The bits in this register and the corresponding $\overline{GPE}$ are cleared by writing a 1 to the corresponding bit location. The status bits in this register do not depend upon the state of a corresponding bit in the general-purpose enable register. See Table 4–18 for a complete description of the register contents.

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	General-purpose event status															
Type	R/C	R	R	R	R/C	R	R	R/C	R	R	R	R/C	R/C	R/C	R/C	R/C
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **General-purpose event status**
Offset: A8h
Type: Read-only, Read/Clear
Default: 0000h

Table 4–18. General-Purpose Event Status Register Description

BIT	SIGNAL	TYPE	FUNCTION
15	ZV_STS	R/C	PC card ZV status. Bit 15 is set on a change in status of bit 6 (ZVENABLE) in the card control register (PCI offset 91h, see Section 4.32).
14–12	RSVD	R	Reserved. Bits 14–12 return 0s when read.
11	PWR_STS	R/C	Power change status. Bit 11 is set when software has changed the power state of the socket. A change in either V_{CC} or V_{PP} for the socket causes this bit to be set.
10–9	RSVD	R	Reserved. Bits 10 and 9 return 0s when read.
8	VPP12_STS	R/C	12-V V_{PP} request status. Bit 8 is set when software has changed the requested V_{PP} level to or from 12 V for the PC Card socket.
7–5	RSVD	R	Reserved. Bits 7–5 return 0s when read.
4	GP4_STS	R/C	GPI4 status. Bit 4 is set on a change in status of the MFUNC5 terminal input level.
3	GP3_STS	R/C	GPI3 status. Bit 3 is set on a change in status of the MFUNC4 terminal input level.
2	GP2_STS	R/C	GPI2 status. Bit 2 is set on a change in status of the MFUNC2 terminal input level.
1	GP1_STS	R/C	GPI1 status. Bit 1 is set on a change in status of the MFUNC1 terminal input level.
0	GP0_STS	R/C	GPI0 status. Bit 0 is set on a change in status of the MFUNC0 terminal input level.

4.44 General-Purpose Event Enable Register

The general-purpose event enable register contains bits that are set to enable a $\overline{GPE}$ signal. The $\overline{GPE}$ signal is driven until the corresponding status bit is cleared and the event is serviced. The $\overline{GPE}$ can be signaled only if one of the multifunction terminals, MFUNC6–MFUNC0, is configured for $\overline{GPE}$ signaling. See Table 4–19 for a complete description of the register contents.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	General-purpose event enable															
Type	R/W	R	R	R	R/W	R	R	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **General-purpose event enable**
Offset: AAh
Type: Read-only, Read/Write
Default: 0000h

Table 4–19. General-Purpose Event Enable Register Description

BIT	SIGNAL	TYPE	FUNCTION
15	ZV_EN	R/W	PC card socket ZV enable. When bit 15 is set, a $\overline{GPE}$ is signaled on a change in status of bit 6 (ZVENABLE) in the card control register (PCI offset 91h, see Section 4.32).
14–12	RSVD	R	Reserved. Bits 14–12 return 0s when read.
11	PWR_EN	R/W	Power change enable. When bit 11 is set, a $\overline{GPE}$ is signaled when software has changed the power state of the socket.
10–9	RSVD	R	Reserved. Bits 10 and 9 return 0s when read.
8	VPP12_EN	R/W	12-V V_{PP} request enable. When bit 8 is set, a $\overline{GPE}$ is signaled when software has changed the requested V_{PP} level to or from 12 V for the card socket.
7–5	RSVD	R	Reserved. Bits 7–5 return 0s when read.
4	GP4_EN	R/W	GPI4 enable. When bit 4 is set, a $\overline{GPE}$ is signaled when there has been a change in status of the MFUNC5 terminal input level if configured as GPI4.
3	GP3_EN	R/W	GPI3 enable. When bit 3 is set, a $\overline{GPE}$ is signaled when there has been a change in status of the MFUNC4 terminal input level if configured as GPI3.
2	GP2_EN	R/W	GPI2 enable. When bit 2 is set, a $\overline{GPE}$ is signaled when there has been a change in status of the MFUNC2 terminal input if configured as GPI2.
1	GP1_EN	R/W	GPI1 enable. When bit 1 is set, a $\overline{GPE}$ is signaled when there has been a change in status of the MFUNC1 terminal input if configured as GPI1.
0	GP0_EN	R/W	GPI0 enable. When bit 0 is set, a $\overline{GPE}$ is signaled when there has been a change in status of the MFUNC0 terminal input if configured as GPI0.

4.45 General-Purpose Input Register

The general-purpose input register provides the logical value of the data input from the GPI terminals, MFUNC5, MFUNC4, and MFUNC2–MFUNC0. See Table 4–20 for a complete description of the register contents.

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	General-purpose input															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	X	X	X	X	X

Register: **General-purpose input**
Offset: ACh
Type: Read-only
Default: 00XXh

Table 4–20. General-Purpose Input Register Description

BIT	SIGNAL	TYPE	FUNCTION
15–5	RSVD	R	Reserved. Bits 15–5 return 0s when read.
4	GPI4_DATA	R	GPI4 data bit. The value read from bit 4 represents the logical value of the data input from the MFUNC5 terminal.
3	GPI3_DATA	R	GPI3 data bit. The value read from bit 3 represents the logical value of the data input from the MFUNC4 terminal.
2	GPI2_DATA	R	GPI2 data bit. The value read from bit 2 represents the logical value of the data input from the MFUNC2 terminal.
1	GPI1_DATA	R	GPI1 data bit. The value read from bit 1 represents the logical value of the data input from the MFUNC1 terminal.
0	GPI0_DATA	R	GPI0 data bit. The value read from bit 0 represents the logical value of the data input from the MFUNC0 terminal.

4.46 General-Purpose Output Register

The general-purpose output register is used for control of the general-purpose outputs. See Table 4–21 for a complete description of the register contents.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	General-purpose output															
Type	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **General-purpose output**
Offset: AEh
Type: Read-only, Read/Write
Default: 0000h

Table 4–21. General-Purpose Output Register Description

BIT	SIGNAL	TYPE	FUNCTION
15–5	RSVD	R	Reserved. Bits 15–5 return 0s when read.
4	GPO4_DATA	R/W	GPO4 data bit. The value written to bit 4 represents the logical value of the data driven to the MFUNC5 terminal if configured as GPO4. Read transactions return the last data value written.
3	GPO3_DATA	R/W	GPIO3 data bit. The value written to bit 3 represents the logical value of the data driven to the MFUNC4 terminal if configured as GPO3. Read transactions return the last data value written.
2	GPO2_DATA	R/W	GPO2 data bit. The value written to bit 2 represents the logical value of the data driven to the MFUNC2 terminal if configured as GPO2. Read transactions return the last data value written.
1	GPO1_DATA	R/W	GPO1 data bit. The value written to bit 1 represents the logical value of the data driven to the MFUNC1 terminal if configured as GPO1. Read transactions return the last data value written.
0	GPO0_DATA	R/W	GPO0 data bit. The value written to bit 0 represents the logical value of the data driven to the MFUNC0 terminal if configured as GPO0. Read transactions return the last data value written.

4.47 Serial Bus Data Register

The serial bus data register is for programmable serial bus byte reads and writes. This register represents the data when generating cycles on the serial bus interface. See Table 4–22 for a complete description of the register contents.

Bit	7	6	5	4	3	2	1	0
Name	Serial bus data							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

Register: **Serial bus data**
Offset: B0h
Type: Read/Write
Default: 00h

Table 4–22. Serial Bus Data Register Description

BIT	SIGNAL	TYPE	FUNCTION
7–0	SBDATA	R/W	Serial bus data. This bit field represents the data byte in a read or write transaction on the serial interface. On reads, bit 5 (REQBUSY) in the serial bus control and status register (PCI offset B3h, see Section 4.50) must be polled to verify that the contents of this register are valid.

4.48 Serial Bus Index Register

The serial bus index register is for programmable serial bus byte reads and writes. This register represents the byte address when generating cycles on the serial bus interface. See Table 4–23 for a complete description of the register contents.

Bit	7	6	5	4	3	2	1	0
Name	Serial bus index							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

Register: **Serial bus index**
Offset: B1h
Type: Read/Write
Default: 00h

Table 4–23. Serial Bus Index Register Description

BIT	SIGNAL	TYPE	FUNCTION
7–0	SBINDEX	R/W	Serial bus index. This bit field represents the byte address in a read or write transaction on the serial interface.

4.49 Serial Bus Slave Address Register

The serial bus slave address register is for programmable serial bus byte read and write transactions. See Table 4–24 for a complete description of the register contents.

Bit	7	6	5	4	3	2	1	0
Name	Serial bus slave address							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

Register: **Serial bus slave address**
Offset: B2h
Type: Read/Write
Default: 00h

Table 4–24. Serial Bus Slave Address Register Description

BIT	SIGNAL	TYPE	FUNCTION
7–1	SLAVADDR	R/W	Serial bus slave address. This bit field represents the slave address of a read or write transaction on the serial interface.
0	RWCMD	R/W	Read/write command. Bit 0 indicates the read/write command bit presented to the serial bus on byte read and write accesses 0 = A byte write access is requested to the serial bus interface. 1 = A byte read access is requested to the serial bus interface.

4.50 Serial Bus Control and Status Register

The serial bus control and status register communicates serial bus status information and selects the quick command protocol. Bit 5 (REQBUSY) in this register must be polled during serial bus byte reads to indicate when data is valid in the serial bus data register. See Table 4–25 for a complete description of the register contents.

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Serial bus control and status							
Type	R/W	R	R	R	R/C	R/W	R/C	R/C
Default	0	0	0	0	0	0	0	0

Register: **Serial bus control and status**
Offset: B3h
Type: Read-only, Read/Write, Read/Clear
Default: 00h

Table 4–25. Serial Bus Control and Status Register Description

BIT	SIGNAL	TYPE	FUNCTION
7	PROT_SEL	R/W	Protocol select. When bit 7 is set, the send byte protocol is used on write requests and the receive byte protocol is used on read commands. The word address byte (SBINDEX) in the serial bus index register (PCI offset B1h, see Section 4.48) is not output by the PCI1410A device when bit 7 is set.
6	RSVD	R	Reserved. Bit 6 returns 0 when read.
5	REQBUSY	R	Requested serial bus access busy. Bit 5 indicates that a requested serial bus access (byte read or write) is in progress. A request is made, and bit 5 is set, by writing to the serial bus slave address register (PCI offset B2h, see Section 4.49). Bit 5 must be polled on reads from the serial bus interface. After the byte read access has been requested, the read data is valid in the serial bus data register.
4	ROMBUSY	R	Serial EEPROM busy status. Bit 4 indicates the status of the PCI1410A serial EEPROM circuitry. Bit 4 is set during the loading of the subsystem ID and other default values from the serial bus EEPROM. 0 = Serial EEPROM circuitry is not busy. 1 = Serial EEPROM circuitry is busy.
3	SBDETECT	R/C	Serial bus detect. Bit 3 is set when the serial bus interface is detected through pullup resistors on the $\overline{\text{VCCD0}}$ and $\overline{\text{VCCD1}}$ terminals after reset. If bit 3 is cleared, the MFUNC4 and MFUNC1 terminals can be used for alternate functions such as general-purpose inputs and outputs. 0 = Serial bus interface is not detected. 1 = Serial bus interface is detected.
2	SBTEST	R/W	Serial bus test. When bit 2 is set, the serial bus clock frequency is increased for test purposes. 0 = Serial bus clock is at normal operating frequency, ≈100 kHz (default). 1 = Serial bus clock frequency is increased for test purposes.
1	REQ_ERR	R/C	Requested serial bus access error. Bit 1 indicates when a data error occurs on the serial interface during a requested cycle and may be set due to a missing acknowledge. Bit 1 is cleared by a write back of 1. 0 = No error is detected during user-requested byte read or write cycle. 1 = Data error is detected during user-requested byte read or write cycle.
0	ROM_ERR	R/C	EEPROM data error status. Bit 0 indicates when a data error occurs on the serial bus interface during the auto-load from the serial bus EEPROM and may be set due to a missing acknowledge. Bit 0 also is set on invalid EEPROM data formats. See Section 3.6.1, *Serial Bus Interface Implementation*, for details on EEPROM data format. Bit 0 is cleared by a write back of 1. 0 = No error is detected during auto-load from serial bus EEPROM. 1 = Data error is detected during auto-load from serial bus EEPROM.

5 ExCA Compatibility Registers

The ExCA registers implemented in the PCI1410A device are register-compatible with the Intel 82365SL–DF PCMCIA controller. The ExCA registers are identified by an offset value that is compatible with the legacy I/O index/data scheme used on the Intel 82365 ISA controller. The ExCA registers are accessed through this scheme by writing the register offset value into the index register (I/O base) and reading or writing the data register (I/O base + 1). The I/O base address used in the index/data scheme is programmed in the PC Card 16-bit I/F legacy mode base address register (PCI offset 44h, see Section 4.28). The offsets from this base address run contiguous from 00h to 3Fh for the socket. See Figure 5–1 for an ExCA I/O-mapping illustration.

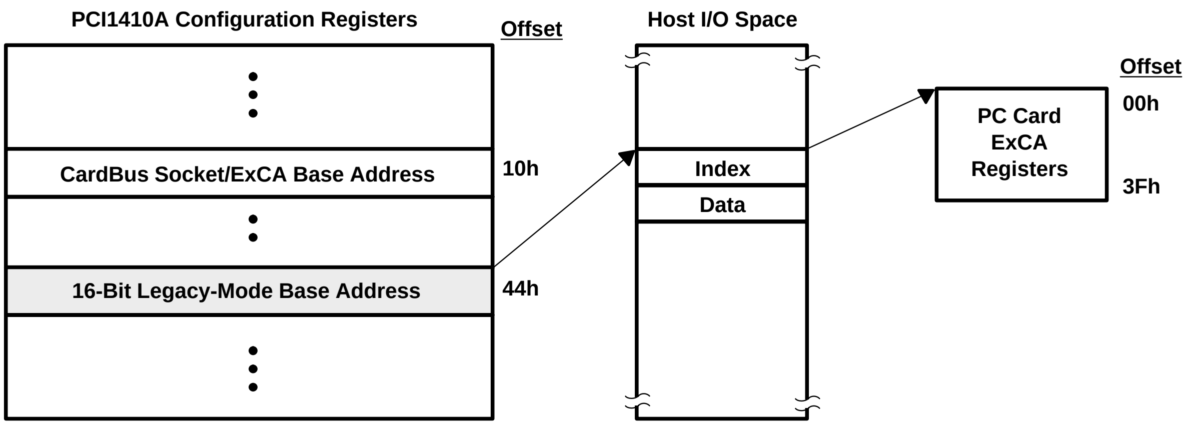


Figure 5–1. ExCA Register Access Through I/O

The TI™ PCI1410A device also provides a memory-mapped alias of the ExCA registers by directly mapping them into PCI memory space. They are located through the CardBus socket/ExCA base address register (PCI offset 10h, see Section 4.12) at memory offset 800h. See Figure 5–2 for an ExCA memory-mapping illustration. This illustration also identifies the CardBus socket-register mapping, which is mapped into the same 4K window at memory offset 0h.

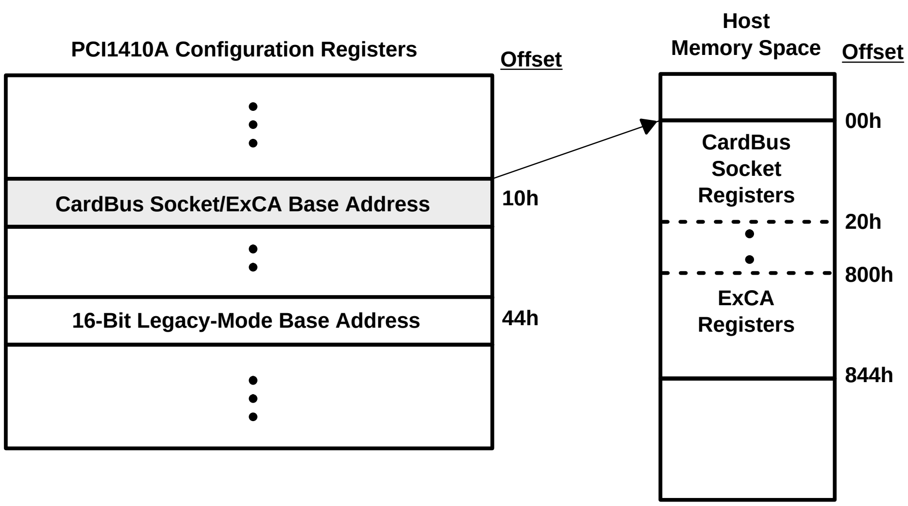


Figure 5–2. ExCA Register Access Through Memory

The interrupt registers in the ExCA register set, as defined by the 82365SL–DL specification, control such card functions as reset, type, interrupt routing, and interrupt enables. Special attention must be paid to the interrupt routing registers and the host-interrupt signaling method selected for the PCI1410A device to ensure that all possible PCI1410A interrupts potentially can be routed to the programmable interrupt controller. The ExCA registers that are critical to the interrupt signaling are the ExCA interrupt and general control register (ExCA offset 03h, see Section 5.4) and the ExCA card status-change-interrupt configuration register (ExCA offset 05h, see Section 5.6).

Access to I/O-mapped 16-bit PC cards is available to the host system via two ExCA I/O windows. These are regions of host I/O address space into which the card I/O space is mapped. These windows are defined by start, end, and offset addresses programmed in the ExCA registers described in this section. I/O windows have byte granularity.

Access to memory-mapped 16-bit PC Cards is available to the host system via five ExCA memory windows. These are regions of host memory space into which the card memory space is mapped. These windows are defined by start, end, and offset addresses programmed in the ExCA registers described in this section. Table 5–1 identifies each ExCA register and its respective ExCA offset. Memory windows have 4-Kbyte granularity.

Table 5–1. ExCA Registers and Offsets

ExCA REGISTER NAME	PCI MEMORY ADDRESS OFFSET (HEX)	ExCA OFFSET (HEX)
Identification and revision	800	00
Interface status	801	01
Power control	802	02
Interrupt and general control	803	03
Card status-change	804	04
Card status-change-interrupt configuration	805	05
Address window enable	806	06
I / O window control	807	07
I / O window 0 start-address low-byte	808	08
I / O window 0 start-address high-byte	809	09
I / O window 0 end-address low-byte	80A	0A
I / O window 0 end-address high-byte	80B	0B
I / O window 1 start-address low-byte	80C	0C
I / O window 1 start-address high-byte	80D	0D
I / O window 1 end-address low-byte	80E	0E
I / O window 1 end-address high-byte	80F	0F
Memory window 0 start-address low-byte	810	10
Memory window 0 start-address high-byte	811	11
Memory window 0 end-address low-byte	812	12
Memory window 0 end-address high-byte	813	13
Memory window 0 offset-address low-byte	814	14
Memory window 0 offset-address high-byte	815	15
Card detect and general control	816	16
Reserved	817	17
Memory window 1 start-address low-byte	818	18
Memory window 1 start-address high-byte	819	19
Memory window 1 end-address low-byte	81A	1A
Memory window 1 end-address high-byte	81B	1B
Memory window 1 offset-address low-byte	81C	1C
Memory window 1 offset-address high-byte	81D	1D

Table 5–1. ExCA Registers and Offsets (Continued)

ExCA REGISTER NAME	PCI MEMORY ADDRESS OFFSET (HEX)	ExCA OFFSET (HEX)
Global control	81E	1E
Reserved	81F	1F
Memory window 2 start-address low-byte	820	20
Memory window 2 start-address high-byte	821	21
Memory window 2 end-address low-byte	822	22
Memory window 2 end-address high-byte	823	23
Memory window 2 offset-address low-byte	824	24
Memory window 2 offset-address high-byte	825	25
Reserved	826	26
Reserved	827	27
Memory window 3 start-address low-byte	828	28
Memory window 3 start-address high-byte	829	29
Memory window 3 end-address low-byte	82A	2A
Memory window 3 end-address high-byte	82B	2B
Memory window 3 offset-address low-byte	82C	2C
Memory window 3 offset-address high-byte	82D	2D
Reserved	82E	2E
Reserved	82F	2F
Memory window 4 start-address low-byte	830	30
Memory window 4 start-address high-byte	831	31
Memory window 4 end-address low-byte	832	32
Memory window 4 end-address high-byte	833	33
Memory window 4 offset-address low-byte	834	34
Memory window 4 offset-address high-byte	835	35
I/O window 0 offset-address low-byte	836	36
I/O window 0 offset-address high-byte	837	37
I/O window 1 offset-address low-byte	838	38
I/O window 1 offset-address high-byte	839	39
Reserved	83A	3A
Reserved	83B	3B
Reserved	83C	3C
Reserved	83D	3D
Reserved	83E	3E
Reserved	83F	3F
Memory window 0 page	840	–
Memory window 1 page	841	–
Memory window 2 page	842	–
Memory window 3 page	843	–
Memory window 4 page	844	–

5.1 ExCA Identification and Revision Register

The ExCA identification and revision register provides host software with information on 16-bit PC Card support and Intel 82365SL-DF compatibility. This register is read-only or read/write, depending on the setting of bit 5 (SUBSYSRW) in the system control register (PCI offset 80h, see Section 4.29). See Table 5–2 for a complete description of the register contents.

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	ExCA identification and revision							
Type	R	R	R/W	R/W	R/W	R/W	R/W	R/W
Default	1	0	0	0	0	1	0	0

Register: **ExCA identification and revision**
Offset: CardBus socket address + 800h; ExCA offset 00h
Type: Read-only, Read/Write
Default: 84h

Table 5–2. ExCA Identification and Revision Register Description

BIT	SIGNAL	TYPE	FUNCTION
7–6	IFTYPE	R	Interface type. These bits, which are hardwired as 10b, identify the 16-bit PC Card support provided by the PCI1410A device. The PCI1410A device supports both I/O and memory 16-bit PC cards.
5–4	RSVD	R/W	Reserved. Bits 5 and 4 can be used for Intel 82365SL-DF emulation.
3–0	365REV	R/W	Intel 82365SL-DF revision. This field stores the Intel 82365SL-DF revision supported by the PCI1410A device. Host software can read this field to determine compatibility to the Intel 82365SL-DF register set. Writing 0010b to this field puts the controller in 82365SL mode. This field defaults to 0100b upon PCI1410A reset.

5.2 ExCA Interface Status Register

The ExCA interface status register provides information on the current status of the PC Card interface. An X in the default bit value indicates that the value of the bit after reset depends on the state of the PC Card interface. See Table 5–3 for a complete description of the register contents.

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	ExCA interface status							
Type	R	R	R	R	R	R	R	R
Default	0	0	X	X	X	X	X	X

Register: **ExCA interface status**
Offset: CardBus socket address + 801h; ExCA offset 01h
Type: Read-only
Default: 00XX XXXXb

Table 5–3. ExCA Interface Status Register Description

BIT	SIGNAL	TYPE	FUNCTION
7	RSVD	R	Reserved. Bit 7 returns 0 when read.
6	CARDPWR	R	Card Power. Bit 6 indicates the current power status of the PC Card socket. This bit reflects how the ExCA power control register (ExCA offset 02h, see Section 5.3) is programmed. Bit 6 is encoded as: 0 = V_{CC} and V_{PP} to the socket are turned off (default). 1 = V_{CC} and V_{PP} to the socket are turned on.
5	READY	R	Ready. Bit 5 indicates the current status of the READY signal at the PC Card interface. 0 = PC Card is not ready for data transfer. 1 = PC Card is ready for data transfer.
4	CARDWP	R	Card write protect. Bit 4 indicates the current status of WP at the PC Card interface. This signal reports to the PCI1410A device whether or not the memory card is write protected. Furthermore, write protection for an entire PCI1410A 16-bit memory window is available by setting bit 7 (WINWP) in the ExCA memory window offset-address high-byte register (see Section 5.18). 0 = WP is 0. PC Card is read/write. 1 = WP is 1. PC Card is read-only.
3	CDETECT2	R	Card detect 2. Bit 3 indicates the status of $\overline{\text{CD2}}$ at the PC Card interface. Software can use this and bit 2 (CDETECT1) to determine if a PC Card is fully seated in the socket. 0 = $\overline{\text{CD2}}$ is 1. No PC Card is inserted. 1 = $\overline{\text{CD2}}$ is 0. PC Card is at least partially inserted.
2	CDETECT1	R	Card detect 1. Bit 2 indicates the status of $\overline{\text{CD1}}$ at the PC Card interface. Software can use this and bit 3 (CDETECT2) to determine if a PC Card is fully seated in the socket. 0 = $\overline{\text{CD1}}$ is 1. No PC Card is inserted. 1 = $\overline{\text{CD1}}$ is 0. PC Card is at least partially inserted.
1–0	BVDSTAT	R	Battery voltage detect. When a 16-bit memory card is inserted, the field indicates the status of the battery voltage detect signals (BVD1, BVD2) at the PC Card interface, where bit 1 reflects the BVD2 status and bit 0 reflects BVD1. 00 = Battery is dead. 01 = Battery is dead. 10 = Battery is low; warning. 11 = Battery is good. When a 16-bit I/O card is inserted, this field indicates the status of $\overline{\text{SPKR}}$ (bit 1) and $\overline{\text{STSCHG}}$ (bit 0) at the PC Card interface. In this case, the two bits in this field directly reflect the current state of these card outputs.

5.3 ExCA Power Control Register

The ExCA power control register provides PC Card power control. Bit 7 (COE) of this register controls the 16-bit output enables on the socket interface, and can be used for power management in 16-bit PC Card applications. The PCI1410A device supports both the 82365SL and 82365SL-DF register models. Bits 3–0 (365REV filed) of the ExCA identification and revision register (ExCA offset 00h, see Section 5.1) control which register model is supported. See Table 5–4 and Table 5–5 for a complete description of the register contents.

5.3.1 Intel 82365SL Support

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	ExCA power control							
Type	R/W	R	R/W	R/W	R	R	R/W	R/W
Default	0	0	0	0	0	0	0	0

Register: **ExCA power control**
Offset: CardBus socket address + 802h; ExCA offset 02h
Type: Read-only, Read/Write
Default: 00h

Table 5–4. ExCA Power Control Register 82365SL Support Description

BIT	SIGNAL	TYPE	FUNCTION
7	COE	R/W	Card output enable. Bit 7 controls the state of all of the 16-bit outputs on the PCI1410A device. This bit is encoded as: 0 = 16-bit PC Card outputs are disabled (default). 1 = 16-bit PC Card outputs are enabled.
6	RSVD	R	Reserved. Bit 6 returns 0 when read.
5	AUTOPWRSWEN	R/W	Auto power switch enable. This bit is enabled by bit 7 of the system control register (PCI offset 80h, see Section 4.29). 0 = Automatic socket power switching based on card detects is disabled. 1 = Automatic socket power switching based on card detects is enabled.
4	CAPWREN	R/W	PC Card power enable. 0 = V_{CC} = V_{PP1} = V_{PP2} = No connection 1 = V_{CC} is enabled and controlled by bit 2 (ExCAPower) of the system control register (PCI offset 80h, see Section 4.29), V_{PP1} and V_{PP2} are controlled according to bits 1–0 (EXCAVPP field).
3–2	RSVD	R	Reserved. Bits 3 and 2 return 0s when read.
1–0	EXCAVPP	R/W	PC Card V_{PP} power control. Bits 1 and 0 are used to request changes to card V_{PP}. The PCI1410A device ignores this field unless V_{CC} to the socket is enabled (that is, 5 V or 3.3 V). This field is encoded as: 00 = No connection (default) 01 = V_{CC} 10 = 12 V 11 = Reserved

5.3.2 Intel 82365SL-DF Support

Bit	7	6	5	4	3	2	1	0
Name	ExCA power control							
Type	R/W	R	R	R/W	R/W	R	R/W	R/W
Default	0	0	0	0	0	0	0	0

Register: **ExCA power control**
Offset: CardBus socket address + 802h; ExCA offset 02h
Type: Read-only, Read/Write
Default: 00h

Table 5–5. ExCA Power Control Register 82365SL-DF Support Description

BIT	SIGNAL	TYPE	FUNCTION
7	COE	R/W	Card output enable. Bit 7 controls the state of all of the 16-bit outputs on the PCI1410A device. This bit is encoded as: 0 = 16-bit PC Card outputs are disabled (default). 1 = 16-bit PC Card outputs are enabled.
6–5	RSVD	R	Reserved. Bits 6 and 5 return 0s when read.
4–3	EXCAVCC	R/W	V_{CC}. Bits 4 and 3 are used to request changes to card V_{CC}. This field is encoded as: 00 = 0 V (default) 01 = 0 V reserved 10 = 5 V 11 = 3.3 V
2	RSVD	R	Reserved. Bit 2 returns 0 when read.
1–0	EXCAVPP	R/W	PC Card V_{PP} power control. Bits 1 and 0 are used to request changes to card V_{PP}. The PCI1410A device ignores this field unless V_{CC} to the socket is enabled (that is, 5 V or 3.3 V). This field is encoded as: 00 = No connection (default) 01 = V_{CC} 10 = 12 V 11 = Reserved

5.4 ExCA Interrupt and General Control Register

The ExCA interrupt and general control register controls interrupt routing for I/O interrupts, as well as other critical 16-bit PC Card functions. See Table 5–6 for a complete description of the register contents.

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	ExCA interrupt and general control							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

Register: **ExCA interrupt and general control**
Offset: CardBus socket address + 803h; ExCA offset 03h
Type: Read/Write
Default: 00h

Table 5–6. ExCA Interrupt and General Control Register Description

BIT	SIGNAL	TYPE	FUNCTION
7	RINGEN	R/W	Card ring indicate enable. Bit 7 enables the ring indicate function of BVD1/$\overline{\text{RI}}$. This bit is encoded as: 0 = Ring indicate is disabled (default). 1 = Ring indicate is enabled.
6	RESET	R/W	Card reset. Bit 6 controls the 16-bit PC Card RESET, and allows host software to force a card reset. Bit 6 affects 16-bit cards only. This bit is encoded as: 0 = RESET signal is asserted (default). 1 = RESET signal is deasserted.
5	CARDTYPE	R/W	Card type. Bit 5 indicates the PC Card type. This bit is encoded as: 0 = Memory PC Card is installed (default). 1 = I/O PC Card is installed.
4	CSCROUTE	R/W	PCI interrupt CSC routing enable bit. When bit 4 is set (high), the card status-change interrupts are routed to PCI interrupts. When low, the card status-change interrupts are routed using bits 7–4 (CSCSELECT field) in the ExCA card status-change interrupt configuration register (ExCA offset 05h, see Section 5.6). This bit is encoded as: 0 = CSC interrupts are routed by ExCA registers (default). 1 = CSC interrupts are routed to PCI interrupts.
3–0	INTSELECT	R/W	Card interrupt select for I/O PC Card functional interrupts. Bits 3–0 select the interrupt routing for I/O PC Card functional interrupts. This field is encoded as: 0000 = No interrupt routing (default). CSC interrupts routed to PCI interrupts. These bit settings are ORed with bit 4 (CSCROUTE) for backwards compatibility. 0001 = IRQ1 enabled 0010 = SMI enabled 0011 = IRQ3 enabled 0100 = IRQ4 enabled 0101 = IRQ5 enabled 0100 = IRQ6 enabled 0111 = IRQ7 enabled 1000 = IRQ8 enabled 1001 = IRQ9 enabled 1010 = IRQ10 enabled 1011 = IRQ11 enabled 1100 = IRQ12 enabled 1101 = IRQ13 enabled 1110 = IRQ14 enabled 1111 = IRQ15 enabled

5.5 ExCA Card Status-Change Register

The ExCA card status-change register controls interrupt routing for I/O interrupts, as well as other critical 16-bit PC Card functions. The register enables these interrupt sources to generate an interrupt to the host. When the interrupt source is disabled, the corresponding bit in this register always reads 0. When an interrupt source is enabled, the corresponding bit in this register is set to indicate that the interrupt source is active. After generating the interrupt to the host, the interrupt service routine must read this register to determine the source of the interrupt. The interrupt service routine is responsible for resetting the bits in this register as well. Resetting a bit is accomplished by one of two methods: a read of this register or an explicit writeback of 1 to the status bit. The choice of these two methods is based on bit 2 (interrupt flag clear mode select) in the ExCA global control register (ExCA offset 1Eh, see Section 5.20). See Table 5–7 for a complete description of the register contents.

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	ExCA card status-change							
Type	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0

Register: **ExCA card status-change**
Offset: CardBus socket address + 804h; ExCA offset 04h
Type: Read-only
Default: 00h

Table 5–7. ExCA Card Status-Change Register Description

BIT	SIGNAL	TYPE	FUNCTION
7–4	RSVD	R	Reserved. Bits 7–4 return 0s when read.
3	CDCHANGE	R	Card detect change. Bit 3 indicates whether a change on $\overline{CD1}$ or $\overline{CD2}$ occurred at the PC Card interface. This bit is encoded as: 0 = No change is detected on either $\overline{CD1}$ or $\overline{CD2}$. 1 = Change is detected on either $\overline{CD1}$ or $\overline{CD2}$.
2	READYCHANGE	R	Ready change. When a 16-bit memory is installed in the socket, bit 2 includes whether the source of a PCI1410A interrupt was due to a change on READY at the PC Card interface, indicating that the PC Card is now ready to accept new data. This bit is encoded as: 0 = No low-to-high transition is detected on READY (default). 1 = Detected low-to-high transition on READY. When a 16-bit I/O card is installed, bit 2 is always 0.
1	BATWARN	R	Battery warning change. When a 16-bit memory card is installed in the socket, bit 1 indicates whether the source of a PCI1410A interrupt was due to a battery-low warning condition. This bit is encoded as: 0 = No battery warning condition (default) 1 = Detected battery warning condition. When a 16-bit I/O card is installed, bit 1 is always 0.
0	BATDEAD	R	Battery dead or status change. When a 16-bit memory card is installed in the socket, bit 0 indicates whether the source of a PCI1410A interrupt was due to a battery-dead condition. This bit is encoded as: 0 = $\overline{STSCHG}$ is deasserted (default). 1 = $\overline{STSCHG}$ is asserted. Ring indicate. When the PCI1410A device is configured for ring-indicate operation, bit 0 indicates the status of $\overline{RI}$.

5.6 ExCA Card Status-Change-Interrupt Configuration Register

The ExCA card status-change-interrupt configuration register controls interrupt routing for card status-change interrupts, as well as masking CSC interrupt sources. See Table 5–8 for a complete description of the register contents.

Bit	7	6	5	4	3	2	1	0
Name	ExCA status-change-interrupt configuration							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

Register: **ExCA card status-change-interrupt configuration**
Offset: CardBus socket address + 805h; ExCA offset 05h
Type: Read/Write
Default: 00h

Table 5–8. ExCA Card Status-Change-Interrupt Configuration Register Description

BIT	SIGNAL	TYPE	FUNCTION
7–4	CSCSELECT	R/W	Interrupt select for card status change. Bits 7–4 select the interrupt routing for card status change interrupts. 0000 = CSC interrupts routed to PCI interrupts if bit 5 (CSC) of the diagnostic register (PCI offset 93h, see Section 4.34) is set to 1. In this case bit 4 (CSCROUTE) of the ExCA interrupt and general control register (ExCA offset 03h, see Section 5.4) is a don't care. This is the default setting. 0000 = No ISA interrupt routing if bit 5 (CSC) of the diagnostic register (PCI offset 93h, see Section 4.34) is set to 0. In this case, CSC interrupts are routed to PCI interrupts by setting bit 4 (CSCROUTE) of the ExCA interrupt and general control register (ExCA offset 03h, see Section 5.4) to 1. This field is encoded as: 0000 = No interrupt routing (default) 1000 = IRQ8 enabled 0001 = IRQ1 enabled 1001 = IRQ9 enabled 0010 = SMI enabled 1010 = IRQ10 enabled 0011 = IRQ3 enabled 1011 = IRQ11 enabled 0100 = IRQ4 enabled 1100 = IRQ12 enabled 0101 = IRQ5 enabled 1101 = IRQ13 enabled 0110 = IRQ6 enabled 1110 = IRQ14 enabled 0111 = IRQ7 enabled 1111 = IRQ15 enabled
3	CDEN	R/W	Card detect enable. Bit 3 enables interrupts on $\overline{CD1}$ or $\overline{CD2}$ changes. This bit is encoded as: 0 = Disables interrupts on $\overline{CD1}$ or $\overline{CD2}$ line changes (default). 1 = Enables interrupts on $\overline{CD1}$ or $\overline{CD2}$ line changes.
2	READYEN	R/W	Ready enable. Bit 2 enables/disables a low-to-high transition on PC Card READY to generate a host interrupt. This interrupt source is considered a card status change. This bit is encoded as: 0 = Disables host interrupt generation (default). 1 = Enables host interrupt generation.
1	BATWARNEN	R/W	Battery warning enable. Bit 1 enables/disables a battery warning condition to generate a CSC interrupt. This bit is encoded as: 0 = Disables host interrupt generation (default). 1 = Enables host interrupt generation.
0	BATDEADEN	R/W	Battery-dead enable. Bit 0 enables/disables a battery-dead condition on a memory PC Card or assertion of the $\overline{STSCHG}$ I/O PC Card signal to generate a CSC interrupt. 0 = Disables host interrupt generation (default). 1 = Enables host interrupt generation.

5.7 ExCA Address Window Enable Register

The ExCA address window enable register enables/disables the memory and I/O windows to the 16-bit PC Card. By default, all windows to the card are disabled. The PCI1410A device does not acknowledge PCI memory or I/O cycles to the card if the corresponding enable bit in this register is 0, regardless of the programming of the ExCA memory or I/O window start/end/offset address registers. See Table 5–9 for a complete description of the register contents.

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	ExCA address window enable							
Type	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

Register: **ExCA address window enable**
Offset: CardBus socket address + 806h; ExCA offset 06h
Type: Read-only, Read/Write
Default: 00h

Table 5–9. ExCA Address Window Enable Register Description

BIT	SIGNAL	TYPE	FUNCTION
7	IOWIN1EN	R/W	I/O window 1 enable. Bit 7 enables/disables I/O window 1 for the PC Card. This bit is encoded as: 0 = I/O window 1 is disabled (default). 1 = I/O window 1 is enabled.
6	IOWIN0EN	R/W	I/O window 0 enable. Bit 6 enables/disables I/O window 0 for the PC Card. This bit is encoded as: 0 = I/O window 0 is disabled (default). 1 = I/O window 0 is enabled.
5	RSVD	R	Reserved. Bit 5 returns 0 when read.
4	MEMWIN4EN	R/W	Memory window 4 enable. Bit 4 enables/disables memory window 4 for the PC Card. This bit is encoded as: 0 = Memory window 4 is disabled (default). 1 = Memory window 4 is enabled.
3	MEMWIN3EN	R/W	Memory window 3 enable. Bit 3 enables/disables memory window 3 for the PC Card. This bit is encoded as: 0 = Memory window 3 is disabled (default). 1 = Memory window 3 is enabled.
2	MEMWIN2EN	R/W	Memory window 2 enable. Bit 2 enables/disables memory window 2 for the PC Card. This bit is encoded as: 0 = Memory window 2 is disabled (default). 1 = Memory window 2 is enabled.
1	MEMWIN1EN	R/W	Memory window 1 enable. Bit 1 enables/disables memory window 1 for the PC Card. This bit is encoded as: 0 = Memory window 1 is disabled (default). 1 = Memory window 1 is enabled.
0	MEMWIN0EN	R/W	Memory window 0 enable. Bit 0 enables/disables memory window 0 for the PC Card. This bit is encoded as: 0 = Memory window 0 is disabled (default). 1 = Memory window 0 is enabled.

5.8 ExCA I/O Window Control Register

The ExCA I/O window control register contains parameters related to I/O window sizing and cycle timing. See Table 5–10 for a complete description of the register contents.

Bit	7	6	5	4	3	2	1	0
Name	ExCA I/O window control							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

Register: **ExCA I/O window control**
Offset: CardBus socket address + 807h; ExCA offset 07h
Type: Read/Write
Default: 00h

Table 5–10. ExCA I/O Window Control Register Description

BIT	SIGNAL	TYPE	FUNCTION
7	WAITSTATE1	R/W	I/O window 1 wait state. Bit 7 controls the I/O window 1 wait state for 16-bit I/O accesses. Bit 7 has no effect on 8-bit accesses. This wait-state timing emulates the ISA wait state used by the Intel 82365SL-DF PCMCIA controller. This bit is encoded as: 0 = 16-bit cycles have standard length (default). 1 = 16-bit cycles are extended by one equivalent ISA wait state.
6	ZEROWS1	R/W	I/O window 1 zero wait state. Bit 6 controls the I/O window 1 wait state for 8-bit I/O accesses. Bit 6 has no effect on 16-bit accesses. This wait-state timing emulates the ISA wait state used by the Intel 82365SL-DF PCMCIA controller. This bit is encoded as: 0 = 8-bit cycles have standard length (default). 1 = 8-bit cycles are reduced to equivalent of three ISA cycles.
5	IOSIS16W1	R/W	I/O window 1 $\overline{\text{IOIS16}}$ source. Bit 5 controls the I/O window 1 automatic data-sizing feature that uses $\overline{\text{IOIS16}}$ from the PC Card to determine the data width of the I/O data transfer. This bit is encoded as: 0 = Window data width is determined by DATASIZE1, bit 4 (default). 1 = Window data width is determined by $\overline{\text{IOIS16}}$.
4	DATASIZE1	R/W	I/O window 1 data size. Bit 4 controls the I/O window 1 data size. Bit 4 is ignored if bit 5 (IOSIS16W1) is set. This bit is encoded as: 0 = Window data width is 8 bits (default). 1 = Window data width is 16 bits.
3	WAITSTATE0	R/W	I/O window 0 wait state. Bit 3 controls the I/O window 0 wait state for 16-bit I/O accesses. Bit 3 has no effect on 8-bit accesses. This wait-state timing emulates the ISA wait state used by the Intel 82365SL-DF PCMCIA controller. This bit is encoded as: 0 = 16-bit cycles have standard length (default). 1 = 16-bit cycles are extended by one equivalent ISA wait state.
2	ZEROWS0	R/W	I/O window 0 zero wait state. Bit 2 controls the I/O window 0 wait state for 8-bit I/O accesses. Bit 2 has no effect on 16-bit accesses. This wait-state timing emulates the ISA wait state used by the Intel 82365SL-DF PCMCIA controller. This bit is encoded as: 0 = 8-bit cycles have standard length (default). 1 = 8-bit cycles are reduced to equivalent of three ISA cycles.
1	IOSIS16W0	R/W	I/O window 0 $\overline{\text{IOIS16}}$ source. Bit 1 controls the I/O window 0 automatic data sizing feature that uses $\overline{\text{IOIS16}}$ from the PC Card to determine the data width of the I/O data transfer. This bit is encoded as: 0 = Window data width is determined by DATASIZE0, bit 0 (default). 1 = Window data width is determined by $\overline{\text{IOIS16}}$.
0	DATASIZE0	R/W	I/O window 0 data size. Bit 0 controls the I/O window 0 data size. Bit 0 is ignored if bit 1 (IOSIS16W0) is set. This bit is encoded as: 0 = Window data width is 8 bits (default). 1 = Window data width is 16 bits.

5.9 ExCA I/O Windows 0 and 1 Start-Address Low-Byte Registers

These registers contain the low byte of the 16-bit I/O window start address for I/O windows 0 and 1. The 8 bits of these registers correspond to the lower 8 bits of the start address.

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	ExCA I/O windows 0 and 1 start-address low-byte							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

Register: **ExCA I/O window 0 start-address low-byte**
Offset: CardBus socket address + 808h; ExCA offset 08h
Register: **ExCA I/O window 1 start-address low-byte**
Offset: CardBus socket address + 80Ch; ExCA offset 0Ch
Type: Read/Write
Default: 00h

5.10 ExCA I/O Windows 0 and 1 Start-Address High-Byte Registers

These registers contain the high byte of the 16-bit I/O window start address for I/O windows 0 and 1. The 8 bits of these registers correspond to the upper 8 bits of the start address.

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	ExCA I/O windows 0 and 1 start-address high-byte							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

Register: **ExCA I/O window 0 start-address high-byte**
Offset: CardBus socket address + 809h; ExCA offset 09h
Register: **ExCA I/O window 1 start-address high-byte**
Offset: CardBus socket address + 80Dh; ExCA offset 0Dh
Type: Read/Write
Default: 00h

5.11 ExCA I/O Windows 0 and 1 End-Address Low-Byte Registers

These registers contain the low byte of the 16-bit I/O window end address for I/O windows 0 and 1. The 8 bits of these registers correspond to the lower 8 bits of the end address.

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	ExCA I/O windows 0 and 1 end-address low-byte							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

Register: **ExCA I/O window 0 end-address low-byte**
Offset: CardBus socket address + 80Ah; ExCA offset 0Ah
Register: **ExCA I/O window 1 end-address low-byte**
Offset: CardBus socket address + 80Eh; ExCA offset 0Eh
Type: Read/Write
Default: 00h

5.12 ExCA I/O Windows 0 and 1 End-Address High-Byte Registers

These registers contain the high byte of the 16-bit I/O window end address for I/O windows 0 and 1. The 8 bits of these registers correspond to the upper 8 bits of the end address.

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	ExCA I/O windows 0 and 1 end-address high-byte							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

Register: **ExCA I/O window 0 end-address high-byte**
Offset: CardBus socket address + 80Bh; ExCA offset 0Bh
Register: **ExCA I/O window 1 end-address high-byte**
Offset: CardBus socket address + 80Fh; ExCA offset 0Fh
Type: Read/Write
Default: 00h

5.13 ExCA Memory Windows 0–4 Start-Address Low-Byte Registers

These registers contain the low byte of the 16-bit memory window start address for memory windows 0, 1, 2, 3, and 4. The 8 bits of these registers correspond to bits A19–A12 of the start address.

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	ExCA memory windows 0–4 start-address low-byte							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

Register: **ExCA memory window 0 start-address low-byte**
Offset: CardBus socket address + 810h; ExCA offset 10h
Register: **ExCA memory window 1 start-address low-byte**
Offset: CardBus socket address + 818h; ExCA offset 18h
Register: **ExCA memory window 2 start-address low-byte**
Offset: CardBus socket address + 820h; ExCA offset 20h
Register: **ExCA memory window 3 start-address low-byte**
Offset: CardBus socket address + 828h; ExCA offset 28h
Register: **ExCA memory window 4 start-address low-byte**
Offset: CardBus socket address + 830h; ExCA offset 30h
Type: Read/Write
Default: 00h

5.14 ExCA Memory Windows 0–4 Start-Address High-Byte Registers

These registers contain the high nibble of the 16-bit memory window start address for memory windows 0, 1, 2, 3, and 4. The lower 4 bits of these registers correspond to bits A23–A20 of the start address. In addition, the memory window data width and wait states are set in this register. See Table 5–11 for a complete description of the register contents.

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	ExCA memory windows 0–4 start-address high byte							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

Register: **ExCA memory window 0 start-address high-byte**
Offset: CardBus socket address + 811h; ExCA offset 11h
Register: **ExCA memory window 1 start-address high-byte**
Offset: CardBus socket address + 819h; ExCA offset 19h
Register: **ExCA memory window 2 start-address high-byte**
Offset: CardBus socket address + 821h; ExCA offset 21h
Register: **ExCA memory window 3 start-address high-byte**
Offset: CardBus socket address + 829h; ExCA offset 29h
Register: **ExCA memory window 4 start-address high-byte**
Offset: CardBus socket address + 831h; ExCA offset 31h
Type: Read/Write
Default: 00h

Table 5–11. ExCA Memory Windows 0–4 Start-Address High-Byte Registers Description

BIT	SIGNAL	TYPE	FUNCTION
7	DATASIZE	R/W	Data size. Bit 7 controls the memory window data width. This bit is encoded as: 0 = Window data width is 8 bits (default). 1 = Window data width is 16 bits.
6	ZEROWAIT	R/W	Zero wait state. Bit 6 controls the memory window wait state for 8- and 16-bit accesses. This wait-state timing emulates the ISA wait state used by the Intel 82365SL-DF PCMCIA controller. This bit is encoded as: 0 = 8- and 16-bit cycles have standard length (default). 1 = 8-bit cycles are reduced to equivalent of three ISA cycles. 16-bit cycles are reduced to equivalent of two ISA cycles.
5–4	SCRATCH	R/W	Scratch-pad bits. Bits 5 and 4 have no effect on memory window operation.
3–0	STAHN	R/W	Start-address high nibble. Bits 3–0 represent the upper address bits A23–A20 of the memory window start address.

5.15 ExCA Memory Windows 0–4 End-Address Low-Byte Registers

These registers contain the low byte of the 16-bit memory window end address for memory windows 0, 1, 2, 3, and 4. The 8 bits of these registers correspond to bits A19–A12 of the end address.

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	ExCA memory windows 0–4 end-address low-byte							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

Register: **ExCA memory window 0 end-address low-byte**
Offset: CardBus socket address + 812h; ExCA offset 12h
Register: **ExCA memory window 1 end-address low-byte**
Offset: CardBus socket address + 81Ah; ExCA offset 1Ah
Register: **ExCA memory window 2 end-address low-byte**
Offset: CardBus socket address + 822h; ExCA offset 22h
Register: **ExCA memory window 3 end-address low-byte**
Offset: CardBus socket address + 82Ah; ExCA offset 2Ah
Register: **ExCA memory window 4 end-address low-byte**
Offset: CardBus socket address + 832h; ExCA offset 32h
Type: Read/Write
Default: 00h

5.16 ExCA Memory Windows 0–4 End-Address High-Byte Registers

These registers contain the high nibble of the 16-bit memory window end address for memory windows 0, 1, 2, 3, and 4. The lower 4 bits of these registers correspond to bits A23–A20 of the end address. In addition, the memory window wait states are set in this register. See Table 5–12 for a complete description of the register contents.

Bit	7	6	5	4	3	2	1	0
Name	ExCA memory windows 0–4 end-address high-byte							
Type	R/W	R/W	R	R	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

Register: **ExCA memory window 0 end-address high-byte**
Offset: CardBus socket address + 813h; ExCA offset 13h
Register: **ExCA memory window 1 end-address high-byte**
Offset: CardBus socket address + 81Bh; ExCA offset 1Bh
Register: **ExCA memory window 2 end-address high-byte**
Offset: CardBus socket address + 823h; ExCA offset 23h
Register: **ExCA memory window 3 end-address high-byte**
Offset: CardBus socket address + 82Bh; ExCA offset 2Bh
Register: **ExCA memory window 4 end-address high-byte**
Offset: CardBus socket address + 833h; ExCA offset 33h
Type: Read-only, Read/Write
Default: 00h

Table 5–12. ExCA Memory Windows 0–4 End-Address High-Byte Registers Description

BIT	SIGNAL	TYPE	FUNCTION
7–6	MEMWS	R/W	Wait state. Bits 7 and 6 specify the number of equivalent ISA wait states to be added to 16-bit memory accesses. The number of wait states added is equal to the binary value of these two bits.
5–4	RSVD	R	Reserved. Bits 5 and 4 return 0s when read.
3–0	ENDHN	R/W	End-address high nibble. Bits 3–0 represent the upper address bits A23–A20 of the memory window end address.

5.17 ExCA Memory Windows 0–4 Offset-Address Low-Byte Registers

These registers contain the low byte of the 16-bit memory window offset address for memory windows 0, 1, 2, 3, and 4. The 8 bits of these registers correspond to bits A19–A12 of the offset address.

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	ExCA memory windows 0–4 offset-address low-byte							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

Register: **ExCA memory window 0 offset-address low-byte**
Offset: CardBus socket address + 814h; ExCA offset 14h
Register: **ExCA memory window 1 offset-address low-byte**
Offset: CardBus socket address + 81Ch; ExCA offset 1Ch
Register: **ExCA memory window 2 offset-address low-byte**
Offset: CardBus socket address + 824h; ExCA offset 24h
Register: **ExCA memory window 3 offset-address low-byte**
Offset: CardBus socket address + 82Ch; ExCA offset 2Ch
Register: **ExCA memory window 4 offset-address low-byte**
Offset: CardBus socket address + 834h; ExCA offset 34h
Type: Read/Write
Default: 00h

5.18 ExCA Memory Windows 0–4 Offset-Address High-Byte Registers

These registers contain the high 6 bits of the 16-bit memory window offset address for memory windows 0, 1, 2, 3, and 4. The lower 6 bits of these registers correspond to bits A25–A20 of the offset address. In addition, the write protection and common/attribute memory configurations are set in this register. See Table 5–13 for a complete description of the register contents.

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	ExCA memory windows 0–4 offset-address high-byte							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

Register: **ExCA memory window 0 offset-address high-byte**
Offset: CardBus socket address + 815h; ExCA offset 15h
Register: **ExCA memory window 1 offset-address high-byte**
Offset: CardBus socket address + 81Dh; ExCA offset 1Dh
Register: **ExCA memory window 2 offset-address high-byte**
Offset: CardBus socket address + 825h; ExCA offset 25h
Register: **ExCA memory window 3 offset-address high-byte**
Offset: CardBus socket address + 82Dh; ExCA offset 2Dh
Register: **ExCA memory window 4 offset-address high-byte**
Offset: CardBus socket address + 835h; ExCA offset 35h
Type: Read/Write
Default: 00h

Table 5–13. ExCA Memory Windows 0–4 Offset-Address High-Byte Registers Description

BIT	SIGNAL	TYPE	FUNCTION
7	WINWP	R/W	Write protect. Bit 7 specifies whether write operations to this memory window are enabled. This bit is encoded as: 0 = Write operations are allowed (default). 1 = Write operations are not allowed.
6	REG	R/W	Bit 6 specifies whether this memory window is mapped to card attribute or common memory. This bit is encoded as: 0 = Memory window is mapped to common memory (default). 1 = Memory window is mapped to attribute memory.
5–0	OFFHB	R/W	Offset-address high byte. Bits 5–0 represent the upper address bits A25–A20 of the memory window offset address.

5.19 ExCA Card Detect and General Control Register

The ExCA card detect and general control register controls how the ExCA registers for the socket respond to card removal, and reports the status of $\overline{VS1}$ and $\overline{VS2}$ at the PC Card interface. See Table 5–14 for a complete description of the register contents.

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	ExCA I/O card detect and general control							
Type	R	R	R/W	R/W	R	R	R/W	R
Default	X	X	0	0	0	0	0	0

Register: **ExCA card detect and general control**
Offset: CardBus socket address + 816h; ExCA offset 16h
Type: Read-only, Read/Write
Default: XX00 0000b

Table 5–14. ExCA Card Detect and General Control Register Description

BIT	SIGNAL	TYPE	FUNCTION
7	VS2STAT	R	$\overline{VS2}$ state. Bit 7 reports the current state of $\overline{VS2}$ at the PC Card interface and, therefore, has no default value. 0 = $\overline{VS2}$ low 1 = $\overline{VS2}$ high
6	VS1STAT	R	$\overline{VS1}$ state. Bit 6 reports the current state of $\overline{VS1}$ at the PC Card interface and, therefore, has no default value. 0 = $\overline{VS1}$ low 1 = $\overline{VS1}$ high
5	SWCSC	R/W	Software card-detect interrupt. If bit 3 (CDEN) in the ExCA card status-change-interrupt configuration register (ExCA offset 05h, see Section 5.6) is set, writing a 1 to bit 5 causes a card-detect card-status-change interrupt for the associated card socket. If bit 3 (CDEN) in the ExCA card status-change-interrupt configuration register (see Section 5.6) is cleared to 0, writing a 1 to bit 5 has no effect. A read operation of this bit always returns 0.
4	CDRESUME	R/W	Card-detect resume enable. If bit 4 is set to 1, once a card-detect change has been detected on $\overline{CD1}$ and $\overline{CD2}$ inputs, $\overline{RI_OUT}$ goes from high to low. $\overline{RI_OUT}$ remains low until bit 0 (card status change) in the ExCA card status-change register (ExCA offset 04h, see Section 5.5) is cleared. If this bit is 0, the card-detect resume functionality is disabled. 0 = Card-detect resume is disabled (default). 1 = Card-detect resume is enabled.
3–2	RSVD	R	Reserved. Bits 3 and 2 return 0s when read.
1	REGCONFIG	R/W	Register configuration on card removal. Bit 1 controls how the ExCA registers for the socket react to a card removal event. This bit is encoded as: 0 = No change to ExCA registers on card removal (default) 1 = Reset ExCA registers on card removal
0	RSVD	R	Reserved. Bit 0 returns 0 when read.

5.20 ExCA Global Control Register

The ExCA global control register controls the PC Card socket. The host interrupt mode bits in this register are retained for Intel 82365SL-DF compatibility. See Table 5–15 for a complete description of the register contents.

Bit	7	6	5	4	3	2	1	0
Name	ExCA global control							
Type	R	R	R	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

Register: **ExCA global control**
Offset: CardBus socket address + 81Eh; ExCA offset 1Eh
Type: Read-only, Read/Write
Default: 00h

Table 5–15. ExCA Global Control Register Description

BIT	SIGNAL	TYPE	FUNCTION
7–5	RSVD	R	Reserved. Bits 7–5 return 0s when read.
4	No function	R/W	This bit has no assigned function.
3	INTMODE	R/W	Level/edge interrupt mode select. Bit 3 selects the signaling mode for the PCI1410A host interrupt. This bit is encoded as: 0 = Host interrupt is edge mode (default). 1 = Host interrupt is level mode.
2	IFCMODE	R/W	Interrupt flag clear mode select. Bit 2 selects the interrupt flag clear mechanism for the flags in the ExCA card status change register (ExCA offset 04h, see Section 5.5). This bit is encoded as: 0 = Interrupt flags are cleared by read of CSC register (default). 1 = Interrupt flags are cleared by explicit writeback of 1.
1	CSCMODE	R/W	Card status change level/edge mode select. Bit 1 selects the signaling mode for the PCI1410A host interrupt for card status changes. This bit is encoded as: 0 = Host interrupt is edge mode (default). 1 = Host interrupt is level mode.
0	PWRDWN	R/W	Power-down mode select. When bit 0 is set to 1, the PCI1410A device is in power-down mode. In power-down mode, the PCI1410A card outputs are high impedance until an active cycle is executed on the card interface. Following an active cycle, the outputs are again high impedance. The PCI1410A device still receives DMA requests, functional interrupts, and/or card status change interrupts; however, an actual card access is required to wake up the interface. This bit is encoded as: 0 = Power-down mode is disabled (default). 1 = Power-down mode is enabled.

5.21 ExCA I/O Windows 0 and 1 Offset-Address Low-Byte Registers

These registers contain the low byte of the 16-bit I/O window offset address for I/O windows 0 and 1. The 8 bits of these registers correspond to the lower 8 bits of the offset address, and bit 0 always is 0.

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	ExCA I/O windows 0 and 1 offset-address low-byte							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R
Default	0	0	0	0	0	0	0	0

Register: **ExCA I/O window 0 offset-address low-byte**
Offset: CardBus socket address + 836h; ExCA offset 36h
Register: **ExCA I/O window 1 offset-address low-byte**
Offset: CardBus socket address + 838h; ExCA offset 38h
Type: Read-only, Read/Write
Default: 00h

5.22 ExCA I/O Windows 0 and 1 Offset-Address High-Byte Registers

These registers contain the high byte of the 16-bit I/O window offset address for I/O windows 0 and 1. The 8 bits of these registers correspond to the upper 8 bits of the offset address.

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	ExCA I/O windows 0 and 1 offset-address high-byte							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

Register: **ExCA I/O window 0 offset-address high-byte**
Offset: CardBus socket address + 837h; ExCA offset 37h
Register: **ExCA I/O window 1 offset-address high-byte**
Offset: CardBus socket address + 839h; ExCA offset 39h
Type: Read/Write
Default: 00h

5.23 ExCA Memory Windows 0–4 Page Register

The upper 8 bits of a 4-byte PCI memory address are compared to the contents of this register when decoding addresses for 16-bit memory windows. Each window has its own page register, all of which default to 00h. By programming this register to a nonzero value, host software can locate 16-bit memory windows in any 1 of 256 16-Mbyte regions in the 4-Gbyte PCI address space. These registers are accessible only when the ExCA registers are memory-mapped; that is, these registers cannot be accessed using the index/data I/O scheme.

Bit	7	6	5	4	3	2	1	0
Name	ExCA memory windows 0–4 page							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

Register: **ExCA memory window 0 page**
Offset: CardBus socket address + 840h
Register: **ExCA memory window 1 page**
Offset: CardBus socket address + 841h
Register: **ExCA memory window 2 page**
Offset: CardBus socket address + 842h
Register: **ExCA memory window 3 page**
Offset: CardBus socket address + 843h
Register: **ExCA memory window 4 page**
Offset: CardBus socket address + 844h
Type: Read/Write
Default: 00h

6 CardBus Socket Registers

The *PC Card Standard* requires a CardBus socket controller to provide five 32-bit registers that report and control socket-specific functions. The PCI1410A device provides the CardBus socket/ExCA base address register (PCI offset 10h, see Section 4.12) to locate these CardBus socket registers in PCI memory address space (see Figure 6–1). Table 6–1 gives the location of the socket registers in relation to the CardBus socket/ExCA base address.

The PCI1410A device implements an additional register at offset 20h that provides power-management control for the socket.

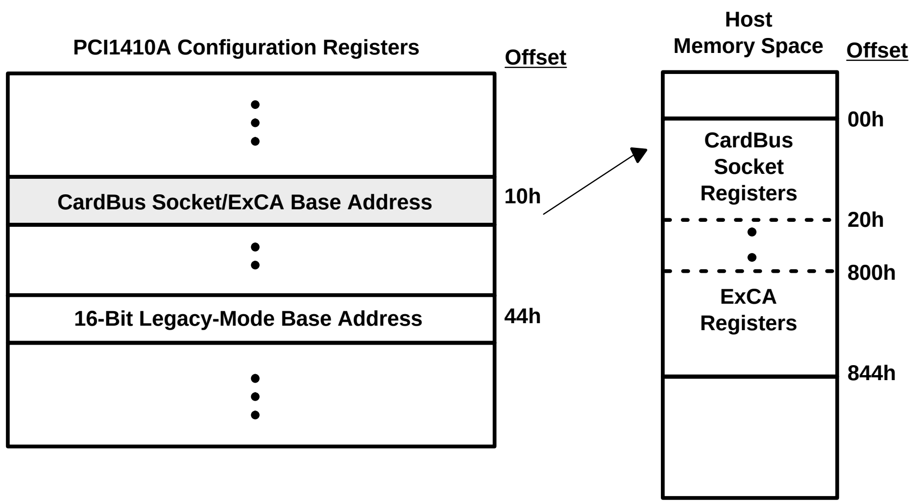


Figure 6–1. Accessing CardBus Socket Registers Through PCI Memory

Table 6–1. CardBus Socket Registers

REGISTER NAME	OFFSET
Socket event	00h
Socket mask	04h
Socket present state	08h
Socket force event	0Ch
Socket control	10h
Reserved	14h
Reserved	18h
Reserved	1Ch
Socket power management	20h

6.1 Socket Event Register

The socket event register indicates a change in socket status has occurred. These bits do not indicate what the change is, only that one has occurred. Software must read the socket present state register (CardBus offset 08h, see Section 6.3) for current status. Each bit in this register can be cleared by writing a 1 to that bit. The bits in this register can be set to a 1 by software by writing a 1 to the corresponding bit in the socket force event register (CardBus offset 0Ch, see Section 6.4). All bits in this register are cleared by PCI reset. They can be set again immediately if, when coming out of PC Card reset, the bridge finds the status unchanged (that is, CSTSCHG is reasserted or card detect still is true). Software must clear this register before enabling interrupts. If it is not cleared when interrupts are enabled, an interrupt is generated (but not masked) based on any bit that is set. See Table 6–2 for a complete description of the register contents.

Bit	**31**	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Name	Socket event															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Socket event															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R/C	R/C	R/C	R/C
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **Socket event**
Offset: CardBus socket address + 00h
Type: Read-only, Read/Clear
Default: 0000 0000h

Table 6–2. Socket Event Register Description

BIT	SIGNAL	TYPE	FUNCTION
31–4	RSVD	R	Reserved. Bits 31–4 return 0s when read.
3	PWREVENT	R/C	Power cycle. Bit 3 is set when the PCI1410A device detects that bit 3 (PWRCYCLE) in the socket present state register (CardBus offset 08h, see Section 6.3) has changed state. This bit is cleared by writing a 1.
2	CD2EVENT	R/C	$\overline{\text{CCD2}}$. Bit 2 is set when the PCI1410A device detects that bit 2 (CDETECT2) in the socket present state register (CardBus offset 08h, see Section 6.3) has changed state. This bit is cleared by writing a 1.
1	CD1EVENT	R/C	$\overline{\text{CCD1}}$. Bit 3 is set when the PCI1410A device detects that bit 1 (CDETECT1) in the socket present state register (CardBus offset 08h, see Section 6.3) has changed state. This bit is cleared by writing a 1.
0	CSTSEVENT	R/C	CSTSCHG. Bit 0 is set when bit 0 (CARDSTS) in the socket present state register (CardBus offset 08h, see Section 6.3) has changed state. For CardBus cards, bit 0 is set on the rising edge of CSTSCHG. For 16-bit PC Cards, bit 0 is set on both transitions of CSTSCHG. This bit is cleared by writing a 1.

6.2 Socket Mask Register

The socket mask register allows software to control the CardBus card events that generate a status change interrupt. The state of these mask bits does not prevent the corresponding bits from reacting in the socket event register (CardBus offset 00h, see Section 6.1). See Table 6–3 for a complete description of the register contents.

Bit	**31**	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Name	Socket mask															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Socket mask															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **Socket mask**
Type: Read-only, Read/Write
Offset: CardBus socket address + 04h
Default: 0000 0000h

Table 6–3. Socket Mask Register Description

BIT	SIGNAL	TYPE	FUNCTION
31–4	RSVD	R	Reserved. Bits 31–4 return 0s when read.
3	PWRMASK	R/W	Power cycle. Bit 3 masks bit 3 (PWRCYCLE) in the socket present state register (CardBus offset 08h, see Section 6.3) from causing a status change interrupt. 0 = PWRCYCLE event does not cause CSC interrupt (default). 1 = PWRCYCLE event causes CSC interrupt.
2–1	CDMASK	R/W	Card detect mask. Bits 2 and 1 mask bits 1 and 2 (CDETECT1 and CDETECT2) in the socket present state register (CardBus offset 08h, see Section 6.3) from causing a CSC interrupt. 00 = Insertion/removal does not cause CSC interrupt (default). 01 = Reserved (undefined) 10 = Reserved (undefined) 11 = Insertion/removal causes CSC interrupt.
0	CSTSMASK	R/W	CSTSCHG mask. Bit 0 masks bit 0 (CARDSTS) in the socket present state register (CardBus offset 08h, see Section 6.3) from causing a CSC interrupt. 0 = CARDSTS event does not cause a CSC interrupt (default). 1 = CARDSTS event causes a CSC interrupt.

6.3 Socket Present State Register

The socket present state register reports information about the socket interface. Write transactions to the socket force event register (CardBus offset 0Ch, see Section 6.4) are reflected here, as well as general socket-interface status. Information about PC Card V_{CC} support and card type is updated only at each insertion. Also, note that the PCI1410A device uses $\overline{\text{CCD1}}$ and $\overline{\text{CCD2}}$ during card identification, and changes on these signals during this operation are not reflected in this register. See Table 6–4 for a complete description of the register contents.

Bit	**31**	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Name	Socket present state															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Socket present state															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	X	0	0	0	X	X	X

Register: **Socket present state**
Offset: CardBus socket address + 08h
Type: Read-only
Default: 3000 00XXh

Table 6–4. Socket Present State Register Description

BIT	SIGNAL	TYPE	FUNCTION
31	YVSOCKET	R	YV socket. Bit 31 indicates whether or not the socket can supply V_{CC} = Y.Y V to PC Cards. The PCI1410A device does not support Y.Y-V V_{CC}; therefore, this bit always is reset unless overridden by the socket force event register (CardBus offset 0Ch, see Section 6.4). This bit is hardwired to 0.
30	XVSOCKET	R	XV socket. Bit 30 indicates whether or not the socket can supply V_{CC} = X.X V to PC Cards. The PCI1410A device does not support X.X-V V_{CC}; therefore, this bit always is reset unless overridden by the socket force event register (CardBus offset 0Ch, see Section 6.4). This bit is hardwired to 0.
29	3VSOCKET	R	3-V socket. Bit 29 indicates whether or not the socket can supply V_{CC} = 3.3 V to PC Cards. The PCI1410A device does support 3.3-V V_{CC}; therefore, this bit always is set unless overridden by the socket force event register (CardBus offset 0Ch, see Section 6.4).
28	5VSOCKET	R	5-V socket. Bit 28 indicates whether or not the socket can supply V_{CC} = 5 V to PC Cards. The PCI1410A device does support 5-V V_{CC}; therefore, this bit always is set unless overridden by the socket force event register (CardBus offset 0Ch, see Section 6.4).
27–14	RSVD	R	Reserved. Bits 27–14 return 0s when read.
13	YVCARD	R	YV card. Bit 13 indicates whether or not the PC Card inserted in the socket supports V_{CC} = Y.Y V.
12	XVCARD	R	XV card. Bit 12 indicates whether or not the PC Card inserted in the socket supports V_{CC} = X.X V.
11	3VCARD	R	3-V card. Bit 11 indicates whether or not the PC Card inserted in the socket supports V_{CC} = 3.3 V.
10	5VCARD	R	5-V card. Bit 10 indicates whether or not the PC Card inserted in the socket supports V_{CC} = 5 V.
9	BADVCCREQ	R	Bad V_{CC} request. Bit 9 indicates that the host software has requested that the socket be powered at an invalid voltage. 0 = Normal operation (default) 1 = Invalid V_{CC} request by host software
8	DATALOST	R	Data lost. Bit 8 indicates that a PC Card removal event may have caused lost data because the cycle did not terminate properly or because write data still resides in the PCI1410A device. 0 = Normal operation (default) 1 = Potential data loss due to card removal
7	NOTACARD	R	Not a card. Bit 7 indicates that an unrecognizable PC Card is inserted in the socket. This bit is not updated until a valid PC Card is inserted into the socket. 0 = Normal operation (default) 1 = Unrecognizable PC Card detected

Table 6–4. Socket Present State Register Description (Continued)

BIT	SIGNAL	TYPE	FUNCTION

6	IREQCINT	R	READY(IREQ)//CINT. Bit 6 indicates the current status of READY(IREQ)//CINT at the PC Card interface.
			0 = READY(IREQ)//CINT low
			1 = READY(IREQ)//CINT high

5	CBCARD	R	CardBus card detected. Bit 5 indicates that a CardBus PC Card is inserted in the socket. This bit is not updated until another card interrogation sequence occurs (card insertion).
4	16BITCARD	R	16-bit card detected. Bit 4 indicates that a 16-bit PC Card is inserted in the socket. This bit is not updated until another card interrogation sequence occurs (card insertion).
3	PWRCYCLE	R	Power cycle. Bit 3 indicates the status of each card powering request. This bit is encoded as: 0 = Socket powered down (default) 1 = Socket powered up
2	CDETECT2	R	CCD2. Bit 2 reflects the current status of CCD2 at the PC Card interface. Changes to this signal during card interrogation are not reflected here. 0 = CCD2 low (PC Card may be present) 1 = CCD2 high (PC Card not present)
1	CDETECT1	R	CCD1. Bit 1 reflects the current status of CCD1 at the PC Card interface. Changes to this signal during card interrogation are not reflected here. 0 = CCD1 low (PC Card may be present) 1 = CCD1 high (PC Card not present)
0	CARDSTS	R	CSTSCHG. Bit 0 reflects the current status of CSTSCHG at the PC Card interface. 0 = CSTSCHG low 1 = CSTSCHG high

6.4 Socket Force Event Register

The socket force event register is used to force changes to the socket event register (CardBus offset 00h, see Section 6.1) and the socket present state register (CardBus offset 08h, see Section 6.3). Bit 14 (CVSTEST) in this register must be written when forcing changes that require card interrogation. See Table 6–5 for a complete description of the register contents.

Bit	**31**	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Name	Socket force event															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Socket force event															
Type	R	W	W	W	W	W	W	W	W	R	W	W	W	W	W	W
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **Socket force event**
Type: Read-only, Write-only
Offset: CardBus socket address + 0Ch
Default: 0000 0000h

Table 6–5. Socket Force Event Register Description

BIT	SIGNAL	TYPE	FUNCTION
31–15	RSVD	R	Reserved. Bits 31–15 return 0s when read.
14	CVSTEST	W	Card VS test. When bit 14 is set, the PCI1410A device reinterrogates the PC Card, updates the socket present state register (CardBus offset 08h, see Section 6.3), and enables the socket control register (CardBus offset 10h, see Section 6.5).
13	FYVCARD	W	Force YV card. Write transactions to bit 13 cause bit 13 (YVCARD) in the socket present state register to be written (CardBus offset 08h, see Section 6.3). When set, this bit disables the socket control register (CardBus offset 10h, see Section 6.5).
12	FXVCARD	W	Force XV card. Write transactions to bit 12 cause bit 12 (XVCARD) in the socket present state register to be written (CardBus offset 08h, see Section 6.3). When set, this bit disables the socket control register (CardBus offset 10h, see Section 6.5).
11	F3VCARD	W	Force 3-V card. Write transactions to bit 11 cause bit 11 (3VCARD) in the socket present state register to be written (CardBus offset 08h, see Section 6.3). When set, this bit disables the socket control register (CardBus offset 10h, see Section 6.5).
10	F5VCARD	W	Force 5-V card. Write transactions to bit 10 cause bit 10 (5VCARD) in the socket present state register to be written (CardBus offset 08h, see Section 6.3). When set, this bit disables the socket control register (CardBus offset 10h, see Section 6.5).
9	FBADVCCREQ	W	Force bad V_{CC} request. Changes to bit 9 (BADVCCREQ) in the socket present state register (CardBus offset 08h, see Section 6.3) can be made by writing to bit 9.
8	FDATALOST	W	Force data lost. Write transactions to bit 8 cause bit 8 (DATALOST) in the socket present state register to be written (CardBus offset 08h, see Section 6.3).
7	FNOTACARD	W	Force not a card. Write transactions to bit 7 cause bit 7 (NOTACARD) in the socket present state register to be written (CardBus offset 08h, see Section 6.3).
6	RSVD	R	Reserved. Bit 6 returns 0 when read.
5	FCBCARD	W	Force CardBus card. Write transactions to bit 5 cause bit 5 (CBCARD) in the socket present state register to be written (CardBus offset 08h, see Section 6.3).
4	F16BITCARD	W	Force 16-bit card. Write transactions to bit 4 cause bit 4 (16BITCARD) in the socket present state register to be written (CardBus offset 08h, see Section 6.3).

Table 6–5. Socket Force Event Register Description (Continued)

BIT	SIGNAL	TYPE	FUNCTION
3	FPWRCYCLE	W	Force power cycle. Write transactions to bit 3 cause bit 3 (PWREVENT) in the socket event register to be written (CardBus offset 00h, see Section 6.1), and bit 3 (PWRCYCLE) in the socket present state register is unaffected (CardBus offset 08h, see Section 6.3).
2	FCDETECT2	W	Force $\overline{\text{CCD2}}$. Write transactions to bit 2 cause bit 2 (CD2EVENT) in the socket event register to be written (CardBus offset 00h, see Section 6.1), and bit 2 (CDETECT2) in the socket present state register is unaffected (CardBus offset 08h, see Section 6.3).
1	FCDETECT1	W	Force $\overline{\text{CCD1}}$. Write transactions to bit 1 cause bit 1 (CD1EVENT) in the socket event register to be written (CardBus offset 00h, see Section 6.1), and bit 1 (CDETECT1) in the socket present state register is unaffected (CardBus offset 08h, see Section 6.3).
0	FCARDSTS	W	Force CSTSCHG. Write transactions to bit 0 cause bit 0 (CSTSEVENT) in the socket event register to be written (CardBus offset 00h, see Section 6.1), and bit 0 (CARDSTS) in the socket present state register is unaffected (CardBus offset 08h, see Section 6.3).

6.5 Socket Control Register

The socket control register provides control of the voltages applied to the socket and instructions for CB $\overline{\text{CLKRUN}}$ protocol. The PCI1410A device ensures that the socket is powered up only at acceptable voltages when a CardBus card is inserted. See Table 6–6 for a complete description of the register contents.

Bit	**31**	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Name	Socket control															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Socket control															
Type	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **Socket control**
Offset: CardBus socket address + 10h
Type: Read-only, Read/Write
Default: 0000 0000h

Table 6–6. Socket Control Register Description

BIT	SIGNAL	TYPE	FUNCTION
31–8	RSVD	R	Reserved. Bits 31–8 return 0s when read.
7	STOPCLK	R/W	CB $\overline{\text{CLKRUN}}$ protocol instructions. 0 = CB $\overline{\text{CLKRUN}}$ protocol can only attempt to stop/slow the CB clock if the socket is idle and the PCI $\overline{\text{CLKRUN}}$ protocol is preparing to stop/slow the PCI bus clock. 1 = CB $\overline{\text{CLKRUN}}$ protocol can attempt to stop/slow the CB clock if the socket is idle.
6–4	VCCCTRL	R/W	V_{CC} control. Bits 6–4 request card V_{CC} changes. 000 = Request power off (default) 001 = Reserved 010 = Request V_{CC} = 5 V 011 = Request V_{CC} = 3.3 V 100 = Request V_{CC} = X.X V 101 = Request V_{CC} = Y.Y V 110 = Reserved 111 = Reserved
3	RSVD	R	Reserved. Bit 3 returns 0 when read.
2–0	VPPCTRL	R/W	V_{PP} control. Bits 2–0 request card V_{PP} changes. 000 = Request power off (default) 001 = Request V_{PP} = 12 V 010 = Request V_{PP} = 5 V 011 = Request V_{PP} = 3.3 V 100 = Request V_{PP} = X.X V 101 = Request V_{PP} = Y.Y V 110 = Reserved 111 = Reserved

6.6 Socket Power Management Register

This register provides power-management control over the socket through a mechanism for slowing or stopping the clock on the card interface when the card is idle. See Table 6–7 for a complete description of the register contents.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Socket power management															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Socket power management															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **Socket power management**
Offset: CardBus socket address + 20h
Type: Read-only, Read/Write
Default: 0000 0000h

Table 6–7. Socket Power Management Register Description

BIT	SIGNAL	TYPE	FUNCTION
31–26	RSVD	R	Reserved. Bits 31–26 return 0s when read.
25	SKTACCES	R	Socket access status. This bit indicates when a socket access has occurred. This bit is cleared by a read access. 0 = A PC card access has not occurred (default). 1 = A PC card access has occurred.
24	SKTMODE	R	Socket mode status. This bit provides clock mode information. 0 = Clock is operating normally. 1 = Clock frequency has changed.
23–17	RSVD	R	Reserved. Bits 23–17 return 0s when read.
16	CLKCTRLEN	R/W	CardBus clock control enable. When bit 16 is set, bit 0 (CLKCTRL) is enabled. 0 = Clock control is disabled (default). 1 = Clock control is enabled.
15–1	RSVD	R	Reserved. Bits 15–1 return 0s when read.
0	CLKCTRL	R/W	CardBus clock control. This bit determines whether the CB $\overline{\text{CLKRUN}}$ protocol stops or slows the CB clock during idle states. Bit 16 (CLKCTRLEN) enables this bit. 0 = Allows CB $\overline{\text{CLKRUN}}$ protocol to stop the CB clock (default). 1 = Allows CB $\overline{\text{CLKRUN}}$ protocol to slow the CB clock by a factor of 16.

7 Distributed DMA (DDMA) Registers

The DMA base address, programmable in PCI configuration space as bits 15–4 (DMABASE field) of the socket DMA register 1 (PCI offset 98h, see Section 4.36), points to a 16-byte region in PCI I/O space where the DMA registers reside. The names and locations of these registers are summarized in Table 7–1. These PCI1410A register definitions are identical in function to, but different in location from, those of the Intel 8237 DMA controller. The similarity between the register models retains some level of compatibility with legacy DMA and simplifies the translation required by the master DMA device when it forwards legacy DMA writes to DMA channels.

While the DMA register definitions are identical to those of the same name in the 8237 DMA controller, some register bits defined in the 8237 DMA controller do not apply to distributed DMA in a PCI environment. In such cases, the PCI1410A device implements these obsolete register bits as nonfunctional, read-only bits. The reserved registers shown in Table 7–1 are implemented as read-only and return 0s when read. Write transactions to reserved registers have no effect.

Table 7–1. Distributed DMA Registers

<table>
<tr><th colspan="4">REGISTER NAME</th><th>DMA BASE ADDRESS OFFSET</th></tr>
<tr><td rowspan="2">Reserved</td><td rowspan="2">Page</td><td colspan="2">Current address</td><td rowspan="2">00h</td></tr>
<tr><td colspan="2">Base address</td></tr>
<tr><td rowspan="2">Reserved</td><td rowspan="2">Reserved</td><td colspan="2">Current count</td><td rowspan="2">04h</td></tr>
<tr><td colspan="2">Base count</td></tr>
<tr><td>N/A</td><td rowspan="2">Reserved</td><td>N/A</td><td>Status</td><td rowspan="2">08h</td></tr>
<tr><td>Mode</td><td>Request</td><td>Command</td></tr>
<tr><td>Multichannel</td><td rowspan="2">Reserved</td><td>N/A</td><td rowspan="2">Reserved</td><td rowspan="2">0Ch</td></tr>
<tr><td>Mask</td><td>Master clear</td></tr>
</table>

7.1 DMA Current Address/Base Address Register

The DMA current address/base address register sets the starting (base) memory address of a DMA transfer. Read transactions from this register indicate the current memory address of a direct memory transfer.

For the 8-bit DMA transfer mode, the current address register contents are presented on AD15–AD0 of the PCI bus during the address phase. Bits 7–0 of the DMA page register (see Section 7.2) are presented on AD23–AD16 of the PCI bus during the address phase.

For the 16-bit DMA transfer mode, the current address register contents are presented on AD16–AD1 of the PCI bus during the address phase, and AD0 is driven to logic 0. Bits 7–1 of the DMA page register (see Section 7.2) are presented on AD23–AD17 of the PCI bus during the address phase, and bit 0 is ignored.

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	DMA current address/base address							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	DMA current address/base address							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

Register: **DMA current address/base address**
Offset: DMA base address + 00h
Type: Read/Write
Default: 0000h

7.2 DMA Page Register

The DMA page register sets the upper byte of the address of a DMA transfer. Details of the address represented by this register are explained in Section 7.1, *DMA Current Address/Base Address Register*.

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	DMA page							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

Register: **DMA page**
Offset: DMA base address + 02h
Type: Read/Write
Default: 00h

7.3 DMA Current Count/Base Count Register

The DMA current count/base count register sets the total transfer count, in bytes, of a direct memory transfer. Read transactions to this register indicate the current count of a direct memory transfer. In the 8-bit transfer mode, the count is decremented by 1 after each transfer, and the count is decremented by 2 after each transfer in the 16-bit transfer mode.

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	DMA current count/base count							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	DMA current count/base count							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

Register: **DMA current count/base count**
Offset: DMA base address + 04h
Type: Read/Write
Default: 0000h

7.4 DMA Command Register

The DMA command register enables and disables the DMA controller. Bit 2 (DMAEN) defaults to 0, enabling the DMA controller. All other bits are reserved. See Table 7–2 for a complete description of the register contents.

Bit	7	6	5	4	3	2	1	0
Name	DMA command							
Type	R	R	R	R	R	R/W	R	R
Default	0	0	0	0	0	0	0	0

Register: **DMA command**
Offset: DMA base address + 08h
Type: Read-only, Read/Write
Default: 00h

Table 7–2. DMA Command Register Description

BIT	SIGNAL	TYPE	FUNCTION
7–3	RSVD	R	Reserved. Bits 7–3 return 0s when read.
2	DMAEN	R/W	DMA controller enable. Bit 2 enables and disables the distributed DMA slave controller in the PCI1410A device and defaults to the enabled state. 0 = DMA controller is enabled (default). 1 = DMA controller is disabled.
1–0	RSVD	R	Reserved. Bits 1 and 0 return 0s when read.

7.5 DMA Status Register

The DMA status register indicates the terminal count and DMA request ($\overline{\text{DREQ}}$) status. See Table 7–3 for a complete description of the register contents.

Bit	7	6	5	4	3	2	1	0
Name	DMA status							
Type	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0

Register: **DMA status**
Offset: DMA base address + 08h
Type: Read-only
Default: 00h

Table 7–3. DMA Status Register Description

BIT	SIGNAL	TYPE	FUNCTION
7–4	DREQSTAT	R	Channel request. In the 8237 DMA controller, bits 7–4 indicate the status of $\overline{\text{DREQ}}$ of each DMA channel. In the PCI1410A device, these bits indicate the $\overline{\text{DREQ}}$ status of the single socket being serviced by this register. All four bits are set to 1 when the PC Card asserts $\overline{\text{DREQ}}$ and are cleared to 0 when $\overline{\text{DREQ}}$ is deasserted. The status of bit 0 (MASKBIT) in the DMA multichannel/mask register (see Section 7.9) has no effect on these bits.
3–0	TC	R	Channel terminal count. The 8237 DMA controller uses bits 3–0 to indicate the TC status of each of its four DMA channels. In the PCI1410A device, these bits report information about a single DMA channel; therefore, all four of these register bits indicate the TC status of the single socket being serviced by this register. All four bits are set to 1 when the TC is reached by the DMA channel. These bits are cleared to 0 when read or the DMA channel is reset.

7.6 DMA Request Register

The DMA request register requests a DMA transfer through software. Any write to this register enables software requests, and this register is to be used in block mode only.

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	DMA request							
Type	W	W	W	W	W	W	W	W
Default	0	0	0	0	0	0	0	0

Register: **DMA request**
Offset: DMA base address + 09h
Type: Write-only
Default: 00h

7.7 DMA Mode Register

The DMA mode register sets the DMA transfer mode. See Table 7–4 for a complete description of the register contents.

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	DMA mode							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R	R
Default	0	0	0	0	0	0	0	0

Register: **DMA mode**
Offset: DMA base address + 0Bh
Type: Read-only, Read/Write
Default: 00h

Table 7–4. DMA Mode Register Description

BIT	SIGNAL	TYPE	FUNCTION
7–6	DMAMODE	R/W	Mode select. The PCI1410A device uses bits 7 and 6 to determine the transfer mode. 00 = Demand mode is selected (default). 01 = Single mode is selected. 10 = Block mode is selected. 11 = Reserved
5	INCDEC	R/W	Address increment/decrement. The PCI1410A device uses bit 5 to select the memory address in the DMA current address/base address register to increment or decrement after each data transfer. This is in accordance with the 8237 DMA controller use of this register bit and is encoded as follows: 0 = Addresses increment (default). 1 = Addresses decrement.
4	AUTOINIT	R/W	Auto initialization 0 = Auto initialization is disabled (default). 1 = Auto initialization is enabled.
3–2	XFERTYPE	R/W	Transfer type. Bits 3 and 2 select the type of direct memory transfer to be performed. A memory write transfer moves data from the PCI1410A PC Card interface to memory and a memory read transfer moves data from memory to the PCI1410A PC Card interface. The field is encoded as: 00 = No transfer selected (default) 01 = Write transfer 10 = Read transfer 11 = Reserved
1–0	RSVD	R	Reserved. Bits 1 and 0 return 0s when read.

7.8 DMA Master Clear Register

The DMA master clear register resets the DMA controller and all DMA registers.

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	DMA master clear							
Type	W	W	W	W	W	W	W	W
Default	0	0	0	0	0	0	0	0

Register: **DMA master clear**
Offset: DMA base address + 0Dh
Type: Write-only
Default: 00h

7.9 DMA Multichannel/Mask Register

The PCI1410A device uses only the least significant bit of this register to mask the PC Card DMA channel. The PCI1410A device sets the mask bit to 1 when the PC Card is removed. Host software is responsible for either resetting the socket's DMA controller or enabling the mask bit. See Table 7–5 for a complete description of the register contents.

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	DMA multichannel/mask							
Type	R	R	R	R	R	R	R	R/W
Default	0	0	0	0	0	0	0	0

Register: **DMA multichannel/mask**
Offset: DMA base address + 0Fh
Type: Read-only, Read/Write
Default: 00h

Table 7–5. DMA Multichannel/Mask Register Description

BIT	SIGNAL	TYPE	FUNCTION
7–1	RSVD	R	Reserved. Bits 7–1 return 0s when read.
0	MASKBIT	R/W	Mask select. Bit 0 masks incoming $\overline{DREQ}$ signals from the PC Card. When set to 1, the socket ignores DMA requests from the card. When cleared (or reset to 0), incoming $\overline{DREQ}$ assertions are serviced normally. 0 = DMA service is provided on card $\overline{DREQ}$. 1 = Socket $\overline{DREQ}$ signal is ignored (default).

8 Electrical Characteristics

8.1 Absolute Maximum Ratings Over Operating Temperature Ranges†

Supply voltage range, V_{CC} . . . . –0.5 V to 4.6 V
Clamping voltage range, V_{CCP}, V_{CCCB}, V_{CCI} . . . . –0.5 V to 6 V
Input voltage range, V_I: PCI . . . . –0.5 V to V_{CCP} + 0.5 V
Card . . . . –0.5 to V_{CCCB} + 0.5 V
Miscellaneous . . . . –0.5 to V_{CCI} + 0.5 V
Fail safe . . . . –0.5 V to V_{CC} + 0.5 V
Output voltage range, V_O: PCI . . . . –0.5 V to V_{CC} + 0.5 V
Card . . . . –0.5 to V_{CC} + 0.5 V
Miscellaneous . . . . –0.5 to V_{CC} + 0.5 V
Fail safe . . . . –0.5 V to V_{CC} + 0.5 V
Input clamp current, I_{IK} ($V_I < 0$ or $V_I > V_{CC}$) (see Note 1) . . . . ±20 mA
Output clamp current, I_{OK} ($V_O < 0$ or $V_O > V_{CC}$) (see Note 2) . . . . ±20 mA
Storage temperature range, T_{stg} . . . . –65°C to 150°C
Virtual junction temperature, T_J . . . . 150°C

† Stresses beyond those listed under absolute maximum ratings may cause permanent damage to the device. These are stress ratings only and functional operation of the device at these or any other conditions beyond those indicated under recommended operating conditions is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

NOTES: 1. Applies for external input and bidirectional buffers. $V_I > V_{CC}$ does not apply to fail-safe terminals. PCI terminals are measured with respect to V_{CCP} instead of V_{CC}. PC Card terminals are measured with respect to V_{CCCB}. Miscellaneous signals are measured with respect to V_{CCI}. The limit specified applies for a dc condition.

2. Applies for external output and bidirectional buffers. $V_O > V_{CC}$ does not apply to fail-safe terminals. PCI terminals are measured with respect to V_{CCP} instead of V_{CC}. PC Card terminals are measured with respect to V_{CCCB}. Miscellaneous signals are measured with respect to V_{CCI}. The limit specified applies for a dc condition.

8.2 Recommended Operating Conditions (see Note 3)

			OPERATION	MIN	NOM	MAX	UNIT
V_{CC}	Core voltage	Commercial	3.3 V	3	3.3	3.6	V
V_{CCP}	PCI I/O clamp voltage	Commercial	3.3 V	3	3.3	3.6	V
			5 V	4.75	5	5.25	
V_{CCCB}	PC Card I/O clamp voltage	Commercial	3.3 V	3	3.3	3.6	V
			5 V	4.75	5	5.25	
V_{CCI}	Miscellaneous I/O clamp voltage	Commercial	3.3 V	3	3.3	3.6	V
			5 V	4.75	5	5.25	
V_{IH}†	High-level input voltage	PCI	3.3 V	0.5 V_{CCP}		V_{CCP}	V
			5 V	2		V_{CCP}	
		PC Card	3.3 V	0.475 V_{CCCB}		V_{CCCB}	
			5 V	2.4		V_{CCCB}	
		Miscellaneous‡		2		V_{CCI}	
		Fail safe§		2		V_{CC}	
		CD terminals‖		2.4		V_{CC}	
V_{IL}†	Low-level input voltage	PCI	3.3 V	0		0.3 V_{CCP}	V
			5 V	0		0.8	
		PC Card	3.3 V	0		0.325 V_{CCCB}	
			5 V	0		0.8	
		Miscellaneous‡		0		0.8	
		Fail safe§		0		0.8	
		CD terminals‖		0		0.75	
V_I	Input voltage	PCI		0		V_{CCP}	V
		PC Card		0		V_{CCCB}	
		Miscellaneous‡		0		V_{CCI}	
		Fail safe§		0		V_{CC}	
V_O¶	Output voltage	PCI		0		V_{CC}	V
		PC Card		0		V_{CC}	
		Miscellaneous‡		0		V_{CC}	
		Fail safe§		0		V_{CC}	
t_t	Input transition time (t_r and t_f)	PCI and PC Card		1		4	ns
		Miscellaneous‡ and fail safe§		0		6	
T_A	Operating ambient temperature range			0	25	70	°C
T_J#	Virtual junction temperature			0	25	115	°C

† Applies to external inputs and bidirectional buffers without hysteresis

‡ Miscellaneous terminals are 70, 62, 59, 60, 61, 64, 65, 67, 68, 69, 73, 74, 71, and 72 for the PGE packaged device; L11, M9, L8, K8, N9, K9, N10, L10, N11, M11, N13, M13, N12, and M12 for the GGU packaged device; and W12, U10, P9, W10, V10, P10, W11, U11, P11, R11, M18, M19, V12, and U12 for the GHK packaged device (SUSPEND, SPKROUT, RI_OUT, multifunction terminals MFUNC0–MFUNC6, and power-switch control terminals).

§ Fail-safe terminals are 75, 117, 131, and 137 for the PGE packaged device; L12, D9, C6, and A4 for the GGU packaged device; and L19, E13, F11, and A9 for the GHK packaged device (card detect and voltage sense terminals).

¶ Applies to external output buffers

These junction temperatures reflect simulation conditions. The customer is responsible for verifying junction temperature.

‖ CD terminals are 75 and 137 for the PGE packaged device; L12 and A4 for the GGU packaged device; and L19 and A9 for the GHK packaged device.

NOTE 3: Unused terminals (input or I/O) must be held high or low to prevent them from floating.

8.3 Electrical Characteristics Over Recommended Operating Conditions (unless otherwise noted)

PARAMETER		TERMINALS	OPERATION	TEST CONDITIONS	MIN	MAX	UNIT
V_{OH}	High-level output voltage (see Note 4)	PCI and Misc1§	3.3 V	I_{OH} = –0.5 mA	$0.9V_{CC}$		V
			5 V	I_{OH} = –2 mA	2.4		
		PC Card	3.3 V	I_{OH} = –0.15 mA	$0.9V_{CC}$		
			5 V	I_{OH} = –0.15 mA	2.4		
		Misc2¶		I_{OH} = –12 mA	2.1		
		Misc3#		I_{OH} = –4 mA	2.1		
V_{OL}	Low-level output voltage	PCI and Misc1§	3.3 V	I_{OL} = 1.5 mA		$0.1V_{CC}$	V
			5 V	I_{OL} = 6 mA		0.55	
		PC Card	3.3 V	I_{OL} = 0.7 mA		$0.1V_{CC}$	
			5 V	I_{OL} = 0.7 mA		0.55	
		Misc2¶		I_{OL} = 12 mA		0.5	
		Misc3#		I_{OL} = 4 mA		0.5	
I_{OZL}	3-state output, high-impedance state output current	Output terminals	3.6 V	$V_I = V_{CC}$		–1	µA
			5.25 V	$V_I = V_{CC}$		–1	
I_{OZH}	3-state output, high-impedance state output current	Output terminals	3.6 V	$V_I = V_{CC}$†		10	µA
			5.25 V	$V_I = V_{CC}$†		25	
I_{IL}	Low-level input current	Input terminals		V_I = GND		–1	µA
		I/O terminals		V_I = GND		–10	
I_{IH}	High-level input current	Input terminals	3.6 V	$V_I = V_{CC}$‡		10	µA
			5.25 V	$V_I = V_{CC}$‡		20	
		I/O terminals	3.6 V	$V_I = V_{CC}$‡		10	
			5.25 V	$V_I = V_{CC}$‡		25	
		Fail-safe terminals	3.6 V	$V_I = V_{CC}$		10	
		V_{CCD} terminals‖	3.6 V	$V_I = V_{CC}$		300	

† For PCI terminals, $V_I = V_{CCP}$. For PC Card terminals, $V_I = V_{CCCB}$. For miscellaneous terminals, $V_I = V_{CCI}$

‡ For I/O terminals, input leakage (I_{IL} and I_{IH}) includes I_{OZ} leakage of the disabled output.

§ Misc1 includes MFUNC6(69), MFUNC5(68), MFUNC4(67), MFUNC3(65), and MFUNC2(64)for the PGE packaged device; M11, N11, L10, N10, and K9 for the GGU packaged device; and R11, P11, U11, W11, and P10 for the GHK packaged device.

¶ Misc2 includes MFUNC1(61), MFUNC0(60), and $\overline{\text{SERR}}$(35) for the PGE packaged device; N9, K8, and M1 for the GGU packaged device; and V10, W10, and R3 for the GHK packaged device.

Misc3 includes SPKROUT(62) and $\overline{\text{RI_OUT}}$(59) for the PGE packaged device; M9 and L8 for the GGU packaged device; and U10 and P9 for the GHK packaged device.

‖ V_{CCD} terminals include $\overline{\text{VCCD0}}$(73) and $\overline{\text{VCCD1}}$(74) for the PGE packaged device; N13 and M13 for the GGU packaged device; and M18 and M19 for the GHK packaged device.

NOTE 4: PCI shared signals are AD31–AD0, C/$\overline{\text{BE3}}$–C/$\overline{\text{BE0}}$, $\overline{\text{FRAME}}$, $\overline{\text{TRDY}}$, $\overline{\text{IRDY}}$, $\overline{\text{STOP}}$, IDSEL, $\overline{\text{DEVSEL}}$, and PAR.

8.4 PCI Clock/Reset Timing Requirements Over Recommended Ranges of Supply Voltage and Operating Free-Air Temperature

PARAMETER		ALTERNATE SYMBOL	TEST CONDITIONS	MIN	MAX	UNIT
t_c	Cycle time, PCLK	t_{cyc}		30		ns
t_{wH}	Pulse duration (width), PCLK high	t_{high}		11		ns
t_{wL}	Pulse duration (width), PCLK low	t_{low}		11		ns
$\Delta v/\Delta t$	Slew rate, PCLK	t_r, t_f		1	4	V/ns
t_w	Pulse duration (width), RSTIN	t_{rst}		1		ms
t_{su}	Setup time, PCLK active at end of $\overline{\text{RSTIN}}$	$t_{rst\text{-}clk}$		100		µs

8.5 PCI Timing Requirements Over Recommended Ranges of Supply Voltage and Operating Free-Air Temperature

This data sheet uses the following conventions to describe time (t) intervals. The format is t_A, where *subscript A* indicates the type of dynamic parameter being represented. The following are used: t_{pd} = propagation delay time, t_d = delay time, t_{su} = setup time, and t_h = hold time.

PARAMETER			ALTERNATE SYMBOL	TEST CONDITIONS	MIN	MAX	UNIT
t_{pd}	Propagation delay time (see Note 4)	PCLK-to-shared signal valid delay time	t_{val}	C_L = 50 pF, See Note 4	2	11	ns
		PCLK-to-shared signal invalid delay time	t_{inv}		2		
t_d	Delay time, t_{en}	Enable time, high impedance-to-active delay time from PCLK	t_{on}		2		ns
	Delay time, t_{dis}	Disable time, active-to-high impedance delay time from PCLK	t_{off}			28	ns
t_{su}	Setup time before PCLK valid		t_{su}		7		ns
t_h	Hold time after PCLK high		t_h		0		ns

NOTE 4: PCI shared signals are AD31–AD0, C/$\overline{\text{BE3}}$–C/$\overline{\text{BE0}}$, $\overline{\text{FRAME}}$, $\overline{\text{TRDY}}$, $\overline{\text{IRDY}}$, $\overline{\text{STOP}}$, IDSEL, $\overline{\text{DEVSEL}}$, and PAR.

8.6 PC Card Cycle Timing

The PC Card cycle timing is controlled by the wait-state bits in the Intel 82365SL-DF compatible memory and I/O window registers. The PC Card cycle generator uses the PCI clock to generate the correct card address setup and hold times and the PC Card command active (low) interval. This allows the cycle generator to output PC Card cycles that are as close to the Intel 82365SL-DF timing as possible, while always slightly exceeding the Intel 82365SL-DF values. This ensures compatibility with existing software and maximizes throughput.

The PC Card address setup and hold times are a function of the wait-state bits. Table 8–1 shows address setup time in PCLK cycles and nanoseconds for I/O and memory cycles. Table 8–2 and Table 8–3 show command active time in PCLK cycles and nanoseconds for I/O and memory cycles. Table 8–4 shows address hold time in PCLK cycles and nanoseconds for I/O and memory cycles.

Table 8–1. PC Card Address Setup Time, $t_{su(A)}$, 8-Bit and 16-Bit PCI Cycles

WAIT-STATE BITS			TS1 – 0 = 01 (PCLK/ns)
I/O			3/90
Memory	WS1	0	2/60
Memory	WS1	1	4/120

Table 8–2. PC Card Command Active Cycle Time, $t_{c(A)}$, 8-Bit PCI Cycles

WAIT-STATE BITS			TS1 – 0 = 01 (PCLK/ns)
	WS	ZWS	
I/O	0	0	19/570
	1	X	23/690
	0	1	7/210
Memory	00	0	19/570
	01	X	23/690
	10	X	23/690
	11	X	23/690
	00	1	7/210

Table 8–3. PC Card Command Active Cycle Time, $t_{c(A)}$, 16-Bit PCI Cycles

WAIT-STATE BITS			TS1 – 0 = 01 (PCLK/ns)
	WS	ZWS	
I/O	0	0	7/210
	1	X	11/330
	0	1	N/A
Memory	00	0	9/270
	01	X	13/390
	10	X	17/510
	11	X	23/630
	00	1	5/150

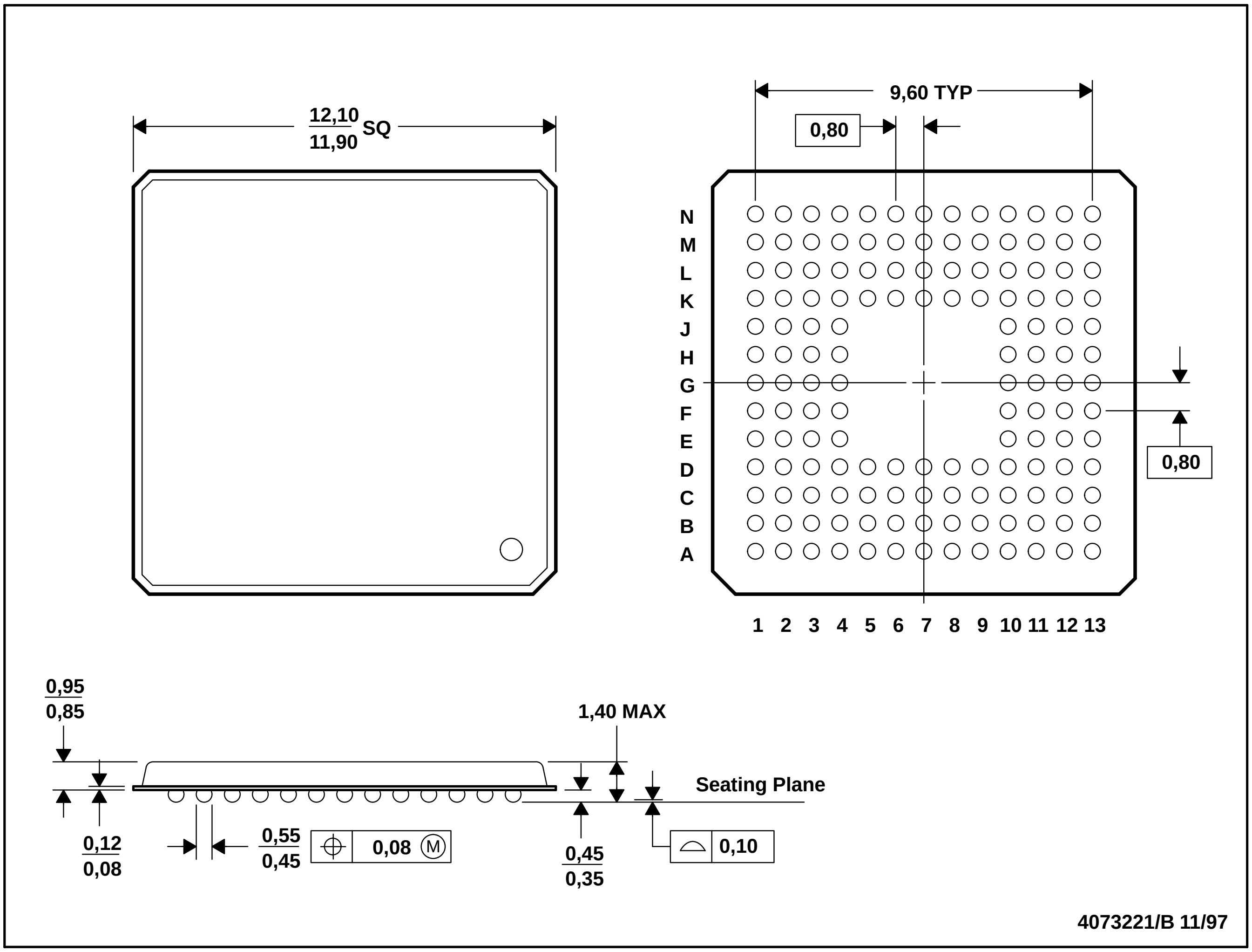
Table 8–4. PC Card Address Hold Time, $t_{h(A)}$, 8-Bit and 16-Bit PCI Cycles

WAIT-STATE BITS			TS1 – 0 = 01 (PCLK/ns)
I/O			2/60
Memory	WS1	0	2/60
Memory	WS1	1	3/90

9 Mechanical Information

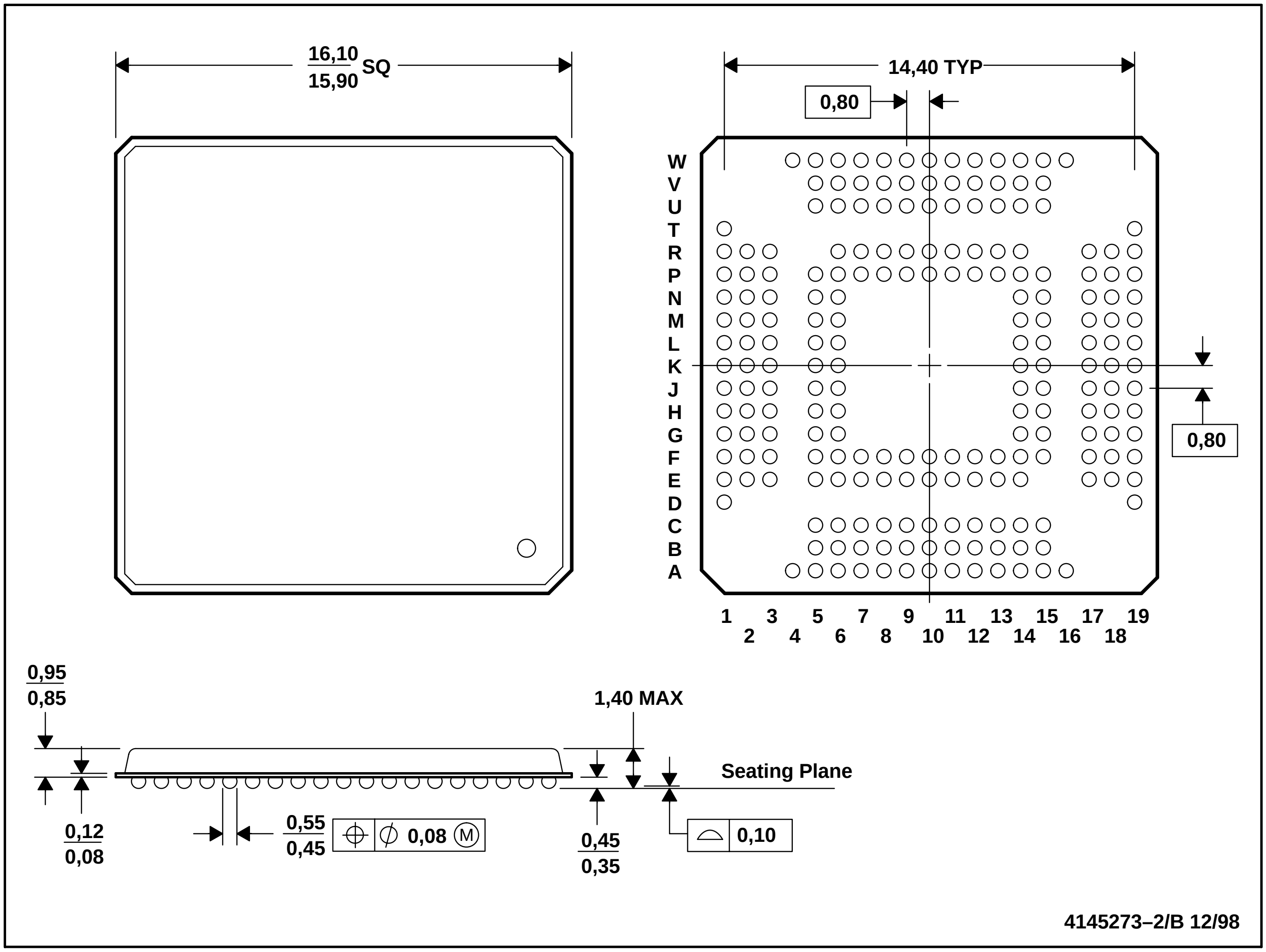
The PCI1410A device is packaged in either a 144-ball GGU MicroStar BGA™ or a 144-pin PGE package. It also is packaged in a 209-ball GHK MicroStar BGA™ that is pin compatible with the TI PCI4410A device. The PCI4410A device is a single-socket CardBus bridge with an integrated OHCI link. The following shows the mechanical dimensions for the GGU, GHK, and PDV packages.

GGU (S-PBGA-N144) **PLASTIC BALL GRID ARRAY**

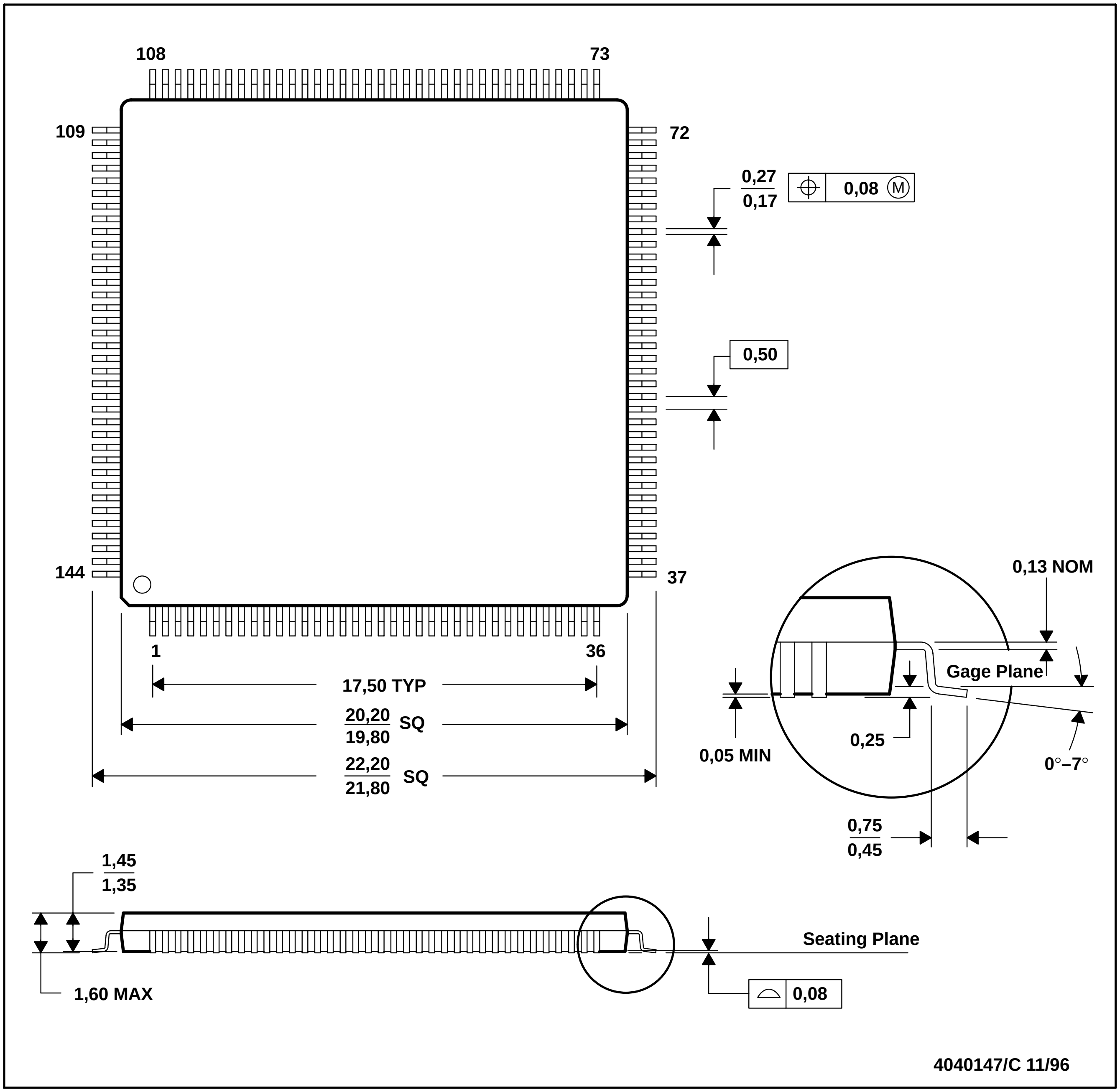


NOTES: A. All linear dimensions are in millimeters.
B. This drawing is subject to change without notice.
C. MicroStar BGA™ configuration

GHK (S-PBGA-N209) PLASTIC BALL GRID ARRAY



NOTES: A. All linear dimensions are in millimeters.
B. This drawing is subject to change without notice.
C. MicroStar BGA™ configuration.



NOTES: A. All linear dimensions are in millimeters.
B. This drawing is subject to change without notice.
C. Falls within JEDEC MS-026